User's Manual

32

RX63N Group, RX631 Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family / RX600 Series

Preliminary

Preliminary Specification

This is a Preliminary specification and is subject to change.

Renesas Electronics
www.renesas.com

Rev.0.90 Dec 2011

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "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.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. 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.
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.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

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

   — 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.

   — 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.

   — 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.

   — 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 one with a different part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

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 RX63N Group, RX631 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	—	—
User's manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX63N Group, RX631 Group User's manual: Hardware	This User's manual
User's manual: Software	Detailed descriptions of the CPU and instruction set	RX Family Series User's manual: Software	REJ09B0435
Application Note	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 xxxxxh

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

Bit	Symbol	Bit Name	Description	R/W
b0	... 0	... Bit	0: ..... 1: Setting prohibited (3)	R/W (1)
b3 to b1	—	Reserved (2)	The read value is 0. The write value should always be 0.	R/W
b4	... 4	... Bit	0: 1:	R
b6, b5	... [1:0]	... Bit	0 0: ..... 0 1: ..... Settings other than above are prohibited. (3)	R/(W)*
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. Make sure to 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 Communication 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

RX63N Group, RX631 Group
Renesas MCUs

R01UH0041EJ0090
Rev.0.90
Dec 12, 2011

100-MHz 32-bit RX MCU, on-chip FPU, 165 DMIPS, up to 2-MB flash memory, Ethernet MAC, full-speed USB 2.0 host/function/OTG interface, various communications interfaces including CAN, 10- & 12-bit A/D converters, RTC

Features

RX63N Group products incorporate an Ethernet controller while RX631 Group products do not.

■ 32-bit RX CPU core
- Max. operating frequency: 100 MHz
- Capable of 165 DMIPS in operation at 100 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 (two-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 500 µA/MHz.
- RTC is capable of operation from a dedicated power supply (min. operating voltage: 2 V).
- Four low-power modes

■ On-chip main flash memory, no wait states
- Supports ROM-less versions and versions with up to 2 Mbytes of ROM (ROM-less version: RX631 Group only)
- 100-MHz operation, 10-ns read cycle (no wait states)
- 384-Kbyte to 2-Mbyte capacities
- User code programmable via the USB, SCI, or JTAG

■ On-chip data flash memory
- ROM-less or 32 Kbytes of ROM (reprogrammable up to 100,000 times)
- Programming/erasing as background operations (BGOs)

■ On-chip SRAM, no wait states
- 32- to 128-Kbyte capacities
- For instructions and operands
- Can provide backup on deep software standby

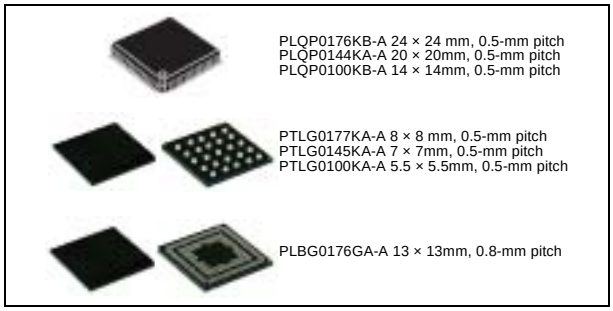
■ DMA
- DMAC: Four channels
- DTC
- EXDMAC: Two channels
- Dedicated DMAC for the Ethernet controller: Single channel

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

■ Clock functions
- External crystal oscillator or internal PLL for operation at 4 to 16 MHz
- Internal 125-kHz LOCO and 50-MHz HOCO
- Dedicated 125-kHz LOCO for the IWDT

■ Real-time clock
- Adjustment functions (30 seconds, leap year, and error)
- Time capture function
  (for capturing times in response to event-signal input on external pins)

■ Independent watchdog timer
- 125-kHz LOCO clock operation



■ Useful functions for IEC60730 compliance
- Oscillation-stoppage detection, frequency measurement, CRC, IWDT, self-diagnostic function for the A/D converter, etc.

■ Various communications interfaces
- Ethernet MAC (1) (not in RX631 Group products)
- Host/function or OTG controller (1) and function controller (1) with full-speed USB 2.0 transfer
- CAN (compliant with ISO11898-1), incorporating 32 mailboxes (up to 3 modules)
- SCI with multiple functionalities (up to 13)
- Choose from among asynchronous mode, clock-synchronous mode, smart-card interface mode, simplified SPI, simplified I2C, and extended serial mode.
  I^2C bus interface for transfer at up to 1 Mbps (up to 4)
- RSPI for high-speed transfer (up to 3)

■ External address space
- Buses for high-speed data transfer (max. operating frequency of 50 MHz)
- 8 CS areas (8 × 16 Mbytes)
- Multiplexed address data or separate address lines are selectable per area.
- 8-, 16-, or 32-bit bus space is selectable per area
- Independent SDRAM area (128 Mbytes)

■ Up to 20 extended-function timers
- 16-bit MTU2: input capture, output compare, PWM waveform output, phase-counting mode (6 channels)
- 16-bit TPU: input capture, output compare, phase-counting mode (12 channels)
- 8-bit TMR (4 channels)
- 16-bit compare-match timers (4 channels)

■ A/D converter for 1-MHz Operation
- Up to 21 12-bit channels, and incorporating 1 sample-and-hold circuit
  Up to 8 10-bit channels, and incorporating 1 sample-and-hold circuit
- Addition of results of A/D conversion (in the 12-bit converter)
- Self diagnosis (for the 10-bit converter)

■ 10-bit D/A converter: 2 channels

■ Temperature sensor for measuring temperature within the chip

■ Register write protection can protect values in important registers against overwriting.

■ Up to 134 pins for GPIO
- 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 is for products with the greatest number of functions, so numbers of peripheral modules and channels will differ in accord with the package. For details, see Table 1.2, Comparison of Functions for Different Packages in the RX63N/RX631 Group.

Table 1.1 Outline of Specifications (1/5)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 100 MHz • 32-bit RX CPU • 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: Nine 32-bit registers Accumulator: One 64-bit register • Basic instructions: 73 • Floating-point instructions: 8 • DSP instructions: 9 • Addressing modes: 10 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32 x 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	ROM	• ROM capacity: 2 Mbytes (max.) • Two on-board programming modes Boot mode (The user area is programmable via the SCI and USB.) User program mode • Parallel programmer mode (for off-board programming)
	RAM	RAM capacity: 128 Kbytes (max.)
	E2 data flash	Data ROM capacity: 32 Kbytes
MCU operating modes		Single-chip mode, on-chip ROM enabled expansion mode, and on-chip ROM disabled expansion mode (software switching)
Clock	Clock generation circuit	• Main clock oscillator, subclock oscillator, low-speed/high-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator • Main-clock oscillation stoppage detection • Separate frequency-division and multiplication settings for the system clock (ICLK), peripheral module clock (PCLK), and external bus clock (BCLK) The CPU and other bus masters run in synchronization with the system clock (ICLK): Up to 100 MHz Peripheral modules run in synchronization with the peripheral module clock (PCLK): Up to 50 MHz Devices connected to the external bus run in synchronization with the external bus clock (BCLK): Up to 50 MHz
Reset		Pin reset, power-on reset, voltage-monitoring reset, independent watchdog timer reset, watchdog timer reset, deep software standby reset, and software reset
Voltage detection circuit		When the voltage on VCC passes the voltage detection level (Vdet), an internal reset or internal interrupt is generated.

Table 1.1 Outline of Specifications (2/5)

Classification	Module/Function	Description
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
Interrupt	Interrupt controller (ICUb)	• Peripheral function interrupts: 187 sources • External interrupts: 16 (pins IRQ0 to IRQ15) • Software interrupts: One source • Non-maskable interrupts: 6 sources Sixteen levels specifiable for the order of priority
External bus extension		• The external address space can be divided into nine areas (CS0 to CS7, SDCS), each with independent control of access settings. Capacity of each area: 16 Mbytes (CS0 to CS7), 128 Mbytes (SDCS) A chip-select signal (CS0# to CS7#, SDCS#) 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 (DMAC)	• 4 channels • Three transfer modes: Normal transfer, repeat transfer, and block transfer • Activation 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 EDACK signal • Capable of direct data transfer to TFT LCD panels • Activation sources: Software trigger, external DMA requests (EDREQ), and interrupt requests from peripheral functions
	Data transfer controller (DTCa)	• Three transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: External interrupts and interrupt requests from peripheral functions
I/O ports	Programmable I/O ports	• I/O ports for the 177-pin TFLGA (in the planning stage), 176-pin LFBGA (in the planning stage), and 176-pin LQFP I/O pins: 133 Input pins: 1 Pull-up resistors: 133 Open-drain outputs: 133 5-V tolerance: 18 • I/O ports for the 145-pin TFLGA (in the planning stage) and 144-pin LQFP I/O pins: 111 Input pins: 1 Pull-up resistors: 111 Open-drain outputs: 111 5-V tolerance: 18 • I/O ports for the 100-pin LQFP I/O pins: 78 Input pins: 1 Pull-up resistors: 78 Open-drain outputs: 78 5-V tolerance: 17

Table 1.1 Outline of Specifications (3/5)

Classification	Module/Function	Description
Timers	16-bit timer pulse unit (TPUa)	• (16 bits x 6 channels) x 2 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 • Buffered operation and phase-counting mode (two phase encoder input) depending on the channel • Support for cascade-connected operation (32 bits x 2 channels) • PPG output trigger can be generated • Capable of generating conversion start triggers for the A/D converters • Signals from the input capture pins are input via a digital filter • Clock frequency measuring method
	Multi-function timer pulse unit 2 (MTU2a)	• (16 bits x 6 channels) x 1 unit • Time bases for the 6 16-bit timer channels can be provided via up to 16 pulse-input/output lines and three pulse-input lines • Select from among eight counter-input clock signals for each channel (PCLK/1, PCLK/4, PCLK/16, PCLK/64, TCLKA, TCLKB, TCLKC, TCLKD) other than channel 5, for which only four signals are available. • Input capture function • 21 output compare/input capture registers • Complementary PWM output mode • Reset synchronous PWM mode • Phase-counting mode • Generation of triggers for A/D converter conversion • Digital filter • Signals from the input capture pins are input via a digital filter • PPG output trigger can be generated • Clock frequency measuring function
	Frequency measuring method (MCK)	The MTU or unit 0 TPU module can be used to monitor the main clock, subclock, HOCO clock, LOCO clock, and PLL clock for abnormal frequencies.
	Port output enable 2 (POE2a)	Controls the high-impedance state of the MTU's waveform output pins
	Programmable pulse generator (PPG)	• (4 bits x 4 groups) x 2 units • Pulse output with the MTU2 or TPU output as a trigger • Maximum of 32 pulse-output possible
	8-bit timers (TMR)	• (8 bits x 2 channels) x 2 units • Select from among seven internal clock signals (PCLK, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/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
	Compare match timer (CMT)	• (16 bits x 2 channels) x 2 units • Select from among four internal clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)
	Realtime clock (RTCa)	• Clock sources: Main clock, subclock • Clock and calendar functions Interrupt sources: Alarm interrupt, periodic interrupt, and carry interrupt • Battery backup operation • Time-capture facility for three values
	Watchdog timer (WDTA)	• 14 bits x 1 channel • Select from among 6 counter-input clock signals (PCLK/4, PCLK/64, PCLK/128, PCLK/512, PCLK/2048, PCLK/8192)
	Independent watchdog timer (IWDTa)	• 14 bits x 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

Table 1.1 Outline of Specifications (4/5)

Classification	Module/Function	Description
Communication function	Ethernet controller (ETHERC)	• 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 Note 1. Magic PacketTM is a registered trademark of Advanced Micro Devices, Inc.
	DMA controller for Ethernet controller (EDMAC)	• Alleviation of CPU loads by the descriptor control method • Transmission FIFO: 2 Kbytes; Reception FIFO: 2 Kbytes
	USB 2.0 host/function module (USBa)	• Includes a UDC (USB Device Controller) and transceiver for USB 2.0 • Host/function module: one port, function module: one port • Compliance with the USB 2.0 specification • Transfer rate: Full speed (12 Mbps) • Self-power mode and bus power are selectable • OTG (On the Go) operation is possible • Incorporates 2 Kbytes of RAM as a transfer buffer
	Serial communications interfaces (SCIc, SCId)	• 13 channels (SCIc: 12 channels + SCId: 1 channel) • SCIc 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 Simple I^2C Simple SPI • SCId (The following functions are added to SCIc) Supports the serial communications protocol, which contains the start frame and information frame Supports the LIN format
	I^2C bus interfaces (RIIC)	• 4 channels (one of them is FM+) • Communication formats I^2C bus format/SMBus format Supports the multi-master Max. transfer rate: 1 Mbps (channel 0)
	IEBus (IEB)	• 1 channel • Supports protocol control for the IEBus Half-duplex asynchronous transfer Multi-master operation Broadcast communications function Two selectable modes, differentiated by transfer rate Note: • IEBus (Inter Equipment Bus) is a registered trademark of Renesas Electronics Corporation.
	CAN module (CAN)	• 3 channels • Compliance with the ISO11898-1 specification (standard frame and extended frame) • 32 mailboxes each
	Serial peripheral interfaces (SPI)	• 3 channels • 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

Table 1.1 Outline of Specifications (5/5)

Classification	Module/Function	Description
12-bit A/D converter (S12ADa)		• 1 unit (1 unit x 14 channels) • 12-bit resolution • Conversion time: 1.0 μs per channel (in operation with PCLK at 50 MHz) • Operating mode Scan mode (single scan mode or continuous scan mode) • Sample-and-hold function • Reference voltage generation • Three ways to start A/D conversion Conversion can be started by software, a conversion start trigger from a timer (MTU, TPU, or TMR), or an external trigger signal. • A/D conversion of the temperature sensor output
10-bit A/D converter (ADb)		• 1 unit (1 unit x 8 channels) • 10-bit resolution • Conversion time: 1.0 μs per channel (in operation with PCLK at 50 MHz) • Operating mode Scan mode (single scan mode or continuous scan mode) External amplifier connection mode • Sample-and-hold function • Three ways to start A/D conversion Conversion can be started by software, a conversion start trigger from a timer (MTU, TPU, or TMR), or an external trigger signal.
D/A converter (DAa)		• 2 channels • 10-bit resolution • Output voltage: 0 V to VREFH
Temperature sensor		• 1 channel • Precision: TBD • The voltage of the temperature is converted into a digital value by the 12-bit A/D converter.
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
Operating frequency		Up to 100 MHz
Power supply voltage		VCC = AVCC0 = VREFH = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0, VBATT = 2.3 V to 3.6 V
Supply current		TBD mA
Operating temperature		−40 to +85°C (products with wide-temperature-range spec.)
Package		177-pin TFLGA (PTLG0177KA-A) (in the planning stage) 176-pin LFBGA (PLBG0176GA-A) (in the planning stage) 176-pin LQFP (PLQP0176KB-A) 145-pin TFLGA (PTLG0145KA-A) (in the planning stage) 144-pin LQFP (PLQP0144KA-A) 100-pin LQFP (PLQP0100KB-A)
On-chip debugging system		• E1 emulator (JTAG and FINE interfaces) • E20 emulator (JTAG interface)

Table 1.2 Comparison of Functions for Different Packages in the RX63N/RX631 Group

Functions		RX63N Group			RX631 Group		
Package		177-pin 176-pin	145-pin 144-pin	100-pin	177-pin 176-pin	145-pin 144-pin	100-pin
External bus width	External bus width	32 bits	16 bits		32 bits	16 bits	
	SDRAM area controller	Available		Not available	Available		Not available
DMA	DMA controller	Ch. 0 to 3			Ch. 0 to 3		
	EXDMA controller	Ch. 0 and 1			Ch. 0 and 1		
	Data transfer controller	Available			Available		
Timers	16-bit timer pulse unit	Ch. 0 to 11		Ch. 0 to 5	Ch. 0 to 11		Ch. 0 to 5
	Multi-function timer pulse unit 2	Ch. 0 to 5			Ch. 0 to 5		
	Port output enable 2	Available			Available		
	Programmable pulse generator	Ch. 0 and 1			Ch. 0 and 1		
	8-bit timers	Ch. 0 to 3			Ch. 0 to 3		
	Compare match timer	Ch. 0 to 3			Ch. 0 to 3		
	Realtime clock	Available			Available		
	Watchdog timer	Available			Available		
	Independent watchdog timer	Available			Available		
Communication function	Ethernet controller	Available			Not available		
	DMA controller for Ethernet controller	Available			Not available		
	USB 2.0 host/function module	Ch. 0 and 1	ch0		Ch. 0 and 1	ch0	
	Serial communications interfaces (SCIc)	Ch. 0 to 11		Ch. 0 to 3, 5, 6, 8 and 9	Ch. 0 to 11		Ch. 0 to 3, 5, 6, 8 and 9
	Serial communications interfaces (SCId)	Ch. 12			Ch. 12		
	I^2C bus interfaces	Ch. 0 to 3		ch0, 2	Ch. 0 to 3		ch0, 2
	IEBus	Available			Available		
	Serial peripheral interfaces	ch0 to 2		Ch. 0 and 1	ch0 to 2		Ch. 0 and 1
	CAN module	For 1.5 M or more: Ch. 0 to 2, For 1 M or less: Ch. 0 and 1		Ch. 0 and 1	For 1.5 M or more: Ch. 0 to 2, For 1 M or less: Ch. 0 and 1		Ch. 0 and 1
12-bit A/D converter (channel)		AN000 to 020		AN000 to 013	AN000 to 020		AN000 to 013
10-bit A/D converter (channel)		AN0 to 7			AN0 to 7		
D/A converter (resolution x channel)		Ch. 0 and 1		ch1	Ch. 0 and 1		ch1
Temperature sensor		Available			Available		
CRC calculator		Available			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	ROM Capacity	RAM Capacity	E²Data Flash	Operating Frequency (Max.)
RX63N	R5F563NACDFP	PLQP0100KB-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NADDFP	PLQP0100KB-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NACDFB	PLQP0144KA-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NADDFB	PLQP0144KA-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NACDLK	PTLG0145KA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NADDLK	PTLG0145KA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NACDFC	PLQP0176KB-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NADDFC	PLQP0176KB-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NACDLC	PTLG0177KA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NADDLC	PTLG0177KA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NACDBG	PLBG0176GA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NADDBG	PLBG0176GA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBCDFP	PLQP0100KB-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBDDFP	PLQP0100KB-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBCDFB	PLQP0144KA-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBDDFB	PLQP0144KA-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBCDLK	PTLG0145KA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBDDLK	PTLG0145KA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBCDFC	PLQP0176KB-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBDDFC	PLQP0176KB-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBCDLC	PTLG0177KA-A-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBDDLC	PTLG0177KA-A-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBCDBG	PLBG0176GA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NBDDBG	PLBG0176GA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDCDFP	PLQP0100KB-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDDDFP	PLQP0100KB-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDCDFB	PLQP0144KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDDDFB	PLQP0144KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDCDLK	PTLG0145KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDDDLK	PTLG0145KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDCDFC	PLQP0176KB-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDDDFC	PLQP0176KB-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDCDLC	PTLG0177KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDDDLC	PTLG0177KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDCDBG	PLBG0176GA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NDDDBG	PLBG0176GA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NECDFP	PLQP0100KB-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NEDDFP	PLQP0100KB-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NECDFB	PLQP0144KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NEDDFB	PLQP0144KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NECDLK	PTLG0145KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz

Table 1.3 List of Products (2/3)

Group	Part No.	Package	ROM Capacity	RAM Capacity	E²Data Flash	Operating Frequency (Max.)
RX63N	R5F563NEDDLK	PTLG0145KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NECDFC	PLQP0176KB-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NEDDFC	PLQP0176KB-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NECDLC	PTLG0177KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NEDDLC	PTLG0177KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NECDBG	PLBG0176GA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F563NEDDBG	PLBG0176GA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
RX631	R5F5631ACDFP	PLQP0100KB-A	768K Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ADDFP	PLQP0100KB-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ACDFB	PLQP0144KA-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ADDFB	PLQP0144KA-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ACDLK	PTLG0145KA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ADDLK	PTLG0145KA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ACDFC	PLQP0176KB-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ADDFC	PLQP0176KB-A	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ACDLC	PTLG0177KA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ADDLC	PTLG0177KA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ACDBG	PLBG0176GA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ADDBG	PLBG0176GA-A*1	768 Kbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BCDFP	PLQP0100KB-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BDDFP	PLQP0100KB-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BCDFB	PLQP0144KA-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BDDFB	PLQP0144KA-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BCDLK	PTLG0145KA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BDDLK	PTLG0145KA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BCDFC	PLQP0176KB-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BDDFC	PLQP0176KB-A	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BCDLC	PTLG0177KA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BDDLC	PTLG0177KA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BCDBG	PLBG0176GA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631BDDBG	PLBG0176GA-A*1	1 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DCDFP	PLQP0100KB-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DDDFP	PLQP0100KB-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DCDFB	PLQP0144KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DDDFB	PLQP0144KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DCDLK	PTLG0145KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DDDLK	PTLG0145KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DCDFC	PLQP0176KB-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DDDFC	PLQP0176KB-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DCDLC	PTLG0177KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DDDLC	PTLG0177KA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DCDBG	PLBG0176GA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631DDDBG	PLBG0176GA-A*1	1.5 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ECDFP	PLQP0100KB-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz

Table 1.3 List of Products (3/3)

Group	Part No.	Package	ROM Capacity	RAM Capacity	E²Data Flash	Operating Frequency (Max.)
RX631	R5F5631EDDFP	PLQP0100KB-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ECDFB	PLQP0144KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631EDDFB	PLQP0144KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ECDLK	PTLG0145KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631EDDLK	PTLG0145KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ECDFC	PLQP0176KB-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631EDDFC	PLQP0176KB-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ECDLC	PTLG0177KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631EDDLC	PTLG0177KA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631ECDBG	PLBG0176GA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5F5631EDDBG	PLBG0176GA-A*1	2 Mbyte	128 Kbytes	32 Kbytes	100 MHz
	R5S56310DCFC	PLQP0176KB-A	0 Mbyte	128 Kbytes	0 Kbytes	100 MHz

Note 1. In the planning stage

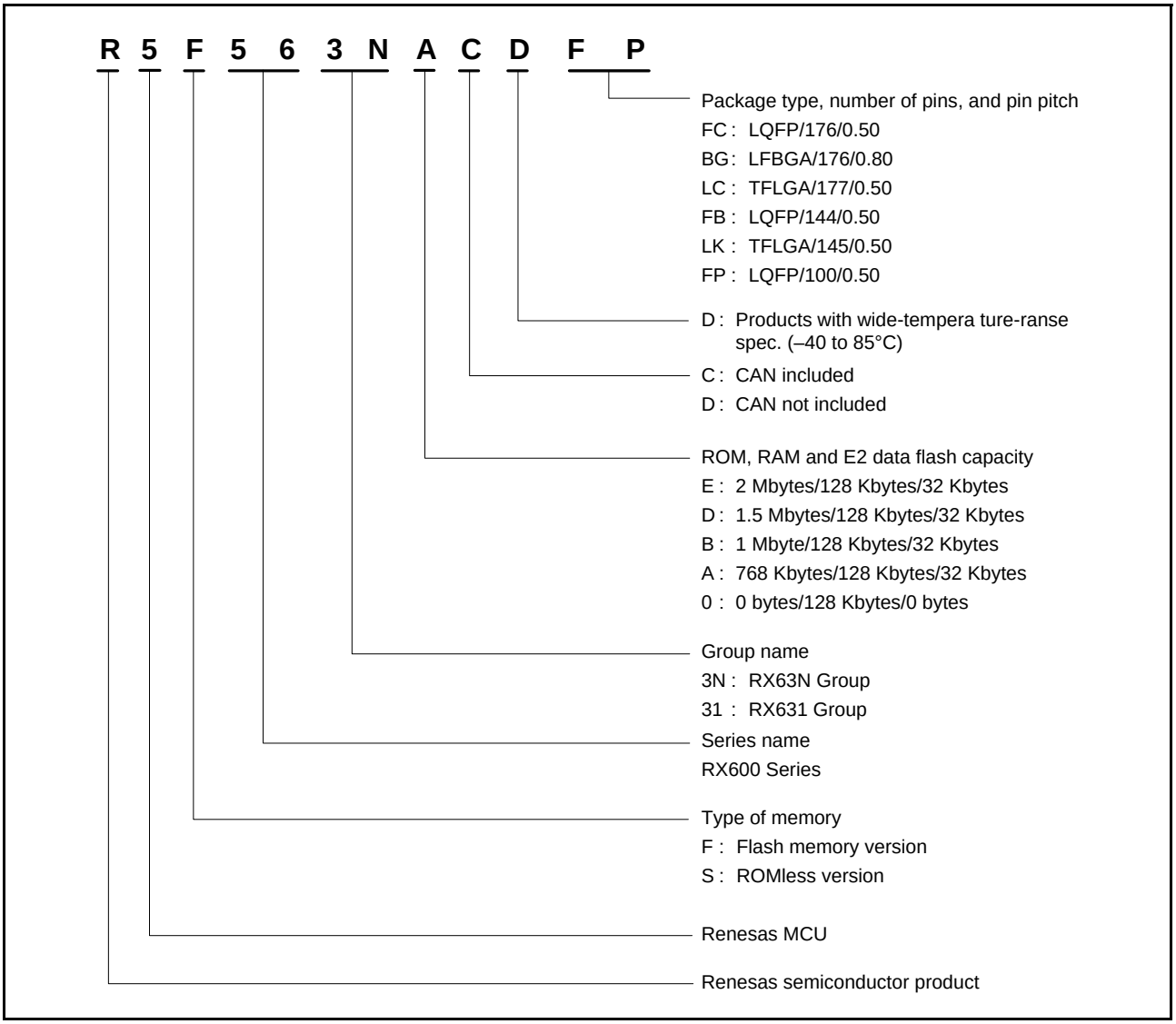
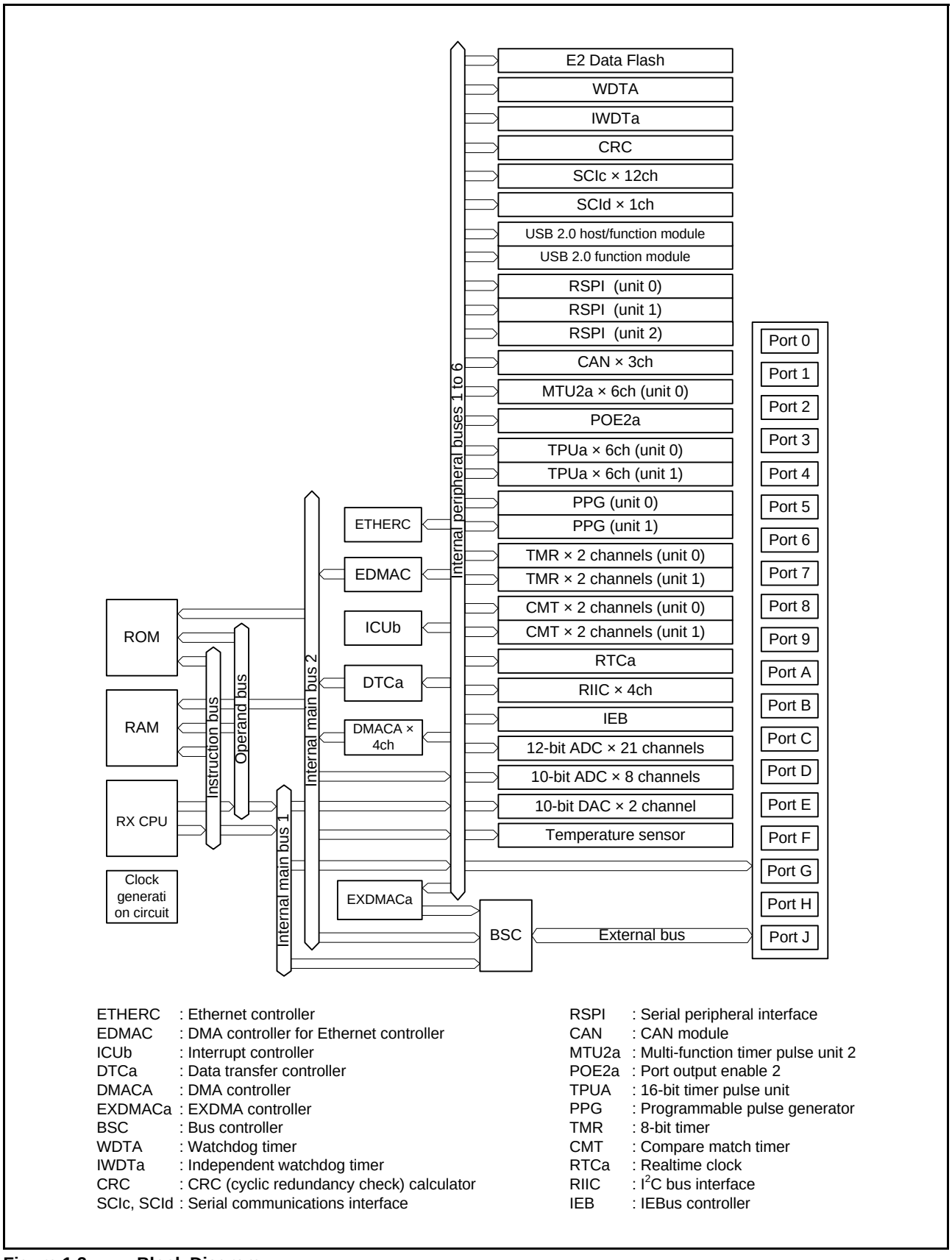


Figure 1.1 How to Read the Product Part No.

1.3 Block Diagram

Figure 1.2 shows a block diagram.



ETHERC : Ethernet controller
EDMAC : DMA controller for Ethernet controller
ICUb : Interrupt controller
DTCa : Data transfer controller
DMACA : DMA controller
EXDMACa : EXDMA controller
BSC : Bus controller
WDTA : Watchdog timer
IWDTa : Independent watchdog timer
CRC : CRC (cyclic redundancy check) calculator
SCIc, SCId : Serial communications interface

RSPI : Serial peripheral interface
CAN : CAN module
MTU2a : Multi-function timer pulse unit 2
POE2a : Port output enable 2
TPUA : 16-bit timer pulse unit
PPG : Programmable pulse generator
TMR : 8-bit timer
CMT : Compare match timer
RTCa : Realtime clock
RIIC : I^2C bus interface
IEB : IEBus controller

Figure 1.2 Block Diagram

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1/6)

Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply.
	VCL	Input	Connect this pin to VSS via a 0.1-μF capacitor. The capacitor should be placed close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
	VBAT	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 clock dedicated for the SDRAM.
	XCOUT	Output	Input/output pins for the subclock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCIN	Input	
Operating mode control	MD	Input	Pins for setting the operating mode. The signal levels on these pins must not be changed during operation.
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	FINEC	Input	Fine interface clock pin
	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 D15	I/O	Input and output pins for the bidirectional data bus.
Multiplexed bus	A0/D0 to A15/D15	I/O	Address/data multiplexed bus
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 and D15 to D8) 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 and D15 to D8) 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.
	CKE	Output	Output pin for SDRAM clock enable signals.
	SDCS#	Output	Output pin for SDRAM chip select signals.
	RAS#	Output	Output pin for SDRAM row address strobe signals.
	CAS#	Output	Output pin for SDRAM column address strobe signals.

Table 1.4 Pin Functions (2/6)

Classifications	Pin Name	I/O	Description
Bus control	WE#	Output	Output pin for SDRAM write enable signals.
	DQM0 to DQM3	Output	Output pins for SDRAM I/O data mask enable signals.
	CS0# to CS7#	Output	Select signals for CS area.
	WAIT#	Input	Input pins for wait request signals in access to the external space.
EXDMA controller	EDREQ0, EDREQ1		Input pins for external DMA transfer requests.
	EDACK0, EDACK1		Output pins for single address transfer acknowledge signals.
Interrupt	NMI	Input	Non-maskable interrupt request signal.
	IRQ0 to IRQ15	Input	Maskable interrupt request signals.
Multi-function timer pulse unit 2	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.
	MTCLKA, MTCLKB MTCLKC, MTCLKD	Input	Input pins for external clock signals.
Port output enable 2	POE0# to POE3# POE8#	Input	Input pins for request signals to place the MTU large-current pins in the high impedance state.
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.
	TIOCA6, TIOCB6 TIOCC6, TIOCD6	I/O	The TGRA6 to TGRD6 input capture input/output compare output/PWM output pins.
	TIOCA7, TIOCB7	I/O	The TGRA7 and TGRB7 input capture input/output compare output/PWM output pins.
	TIOCA8, TIOCB8	I/O	The TGRA8 and TGRB8 input capture input/output compare output/PWM output pins.
	TIOCA9, TIOCB9 TIOCC9, TIOCD9	I/O	The TGRA9 to TGRD9 input capture input/output compare output/PWM output pins.
	TIOCA10, TIOCB10	I/O	The TGRA10 and TGRB10 input capture input/output compare output/PWM output pins.
	TIOCA11, TIOCB11	I/O	The TGRA11 and TGRB11 input capture input/output compare output/PWM output pins.
	TCLKE, TCLKF TCLKG, TCLKH	Input	Input pins for external clock signals.
Programmable pulse generator	PO0 to PO31	Output	Output pins for the pulse signals.

Table 1.4 Pin Functions (3/6)

Classifications	Pin Name	I/O	Description
8-bit timer	TMO0 to TMO3	Output	Output pins for the compare match signals.
	TMCI0 to TMCI3	Input	Input pins for the external clock signals that drive for the counters.
	TMRI0 to TMRI3	Input	Input pins for the counter-reset signals.
Serial communications interface (SCIc)	• Asynchronous mode/clock synchronous mode		
	SCK0 to SCK11	I/O	Input/output pins for clock signals.
	RXD0 to RXD11	Input	Input pins for data reception.
	TXD0 to TXD11	Output	Output pins for data transmission.
	CTS0# to CTS11#	Input	Transfer start control input pins
	RTS0# to RTS11#	Output	Transfer start control output pins
	• Simple I^2C mode		
	SSCL0 to SSCL11	I/O	Input/output pins for the I^2C clock
	SSDA0 to SSDA11	I/O	Input/output pins for the I^2C data
Serial communications interface (SCIc)	• Simple SPI mode		
	SCK0 to SCK11	I/O	Input/output pins for the clock
	SMISO0 to SMISO11	I/O	Input/output pins for slave transmit data.
	SMOSI0 to SMOSI11	I/O	Input/output pins for master transmit data.
	SS0# to SS11#	Input	Input pins for chip select signals
Serial communications interface (SCId)	• Asynchronous mode/clock synchronous mode		
	SCK12	I/O	Input/output pin for clock signals.
	RXD12	Input	Input pin for data reception.
	TXD12	Output	Output pin for data transmission.
	CTS12#	Input	Transfer start control input pins
	RTS12#	Output	Transfer start control output pins
	• Simple I^2C mode		
	SSCL12	I/O	Input/output pins for the I^2C clock
	SSDA12	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK12	I/O	Input/output pins for the clock
	SMISO12	I/O	Input/output pins for slave transmit data.
	SMOSI12	I/O	Input/output pins for master transmit data.
	SS12#	Input	Input pins for chip select signals
	• Extended serial mode		
	RXDX12	Input	Input pin for receive data
	TXDX12	Output	Output pin for transmit data
	SIO12	I/O	Input/output pin for transfer data
I^2C bus interface	SCL0[FM+], SCL1 to SCL3	I/O	Input/output pin for clocks. Bus can be directly driven by the N-channel open drain output.
	SDA0[FM+], SDA1 to SDA3	I/O	Input/output pin for data. Bus can be directly driven by the N-channel open drain output.

Table 1.4 Pin Functions (4/6)

Classifications	Pin Name	I/O	Description
Ethernet controller	REF50CK	Input	50-MHz reference clock. This pin inputs reference signals for transmission/reception timings in RMII mode.
	RMII_CRS_DV	Input	Indicates that there are carrier detection signals and valid receive data on RMII_RXD1 and RMII_RXD0 in RMII mode.
	RMII_TXD0, RMII_TXD1	Output	2-bit transmit data in RMII mode.
	RMII_RXD0, RMII_RXD1	Input	2-bit receive data in RMII mode.
	RMII_TXD_EN	Output	Output pin for data transmit enable signals in RMII mode.
	RMII_RX_ER	Input	Indicates an error has occurred during reception of data in RMII mode.
	ET_CRS	Input	Carrier detection/data reception enable pin.
	ET_RX_DV	Input	Indicates that there are valid receive data on ET_ERXD3 to ET_ERXD0.
	ET_EXOUT	Output	General-purpose external output pin.
	ET_LINKSTA	Input	Inputs link status from the PHY-LSI.
	ET_ETXD0 to ET_ETXD3	Output	4 bits of MII transmit data.
	ET_ERXD0 to ET_ERXD3	Input	4 bits of MII receive data.
	ET_TX_EN	Output	Transmit enable pin. Indicates that transmit data is ready on ET_ETXD3 to ET_ETXD0.
	ET_TX_ER	Output	Transmit error pin. Notifies the PHY_LSI of an error during transmission.
	ET_RX_ER	Input	Receive error pin. Recognizes an error during reception.
	ET_TX_CLK	Input	Transmit clock pin. This pin inputs reference signals for output timings from ET_TX_EN, ET_ETXD3 to ET_ETXD0, and ET_TX_ER.
	ET_RX_CLK	Input	Receive clock pin. This pin inputs reference signals for input timings to ET_RX_DV, ET_ERXD3 to ET_ERXD0, and ET_RX_ER.
Ethernet controller	ET_COL	Input	Inputs collision detection signals.
	ET_WOL	Output	Receives Magic packets.
	ET_MDC	Output	Outputs reference clock signals for information transfer via ET_MDIO.
	ET_MDIO	I/O	Inputs or outputs bidirectional signals for exchange of management information between the RX63N Group and the PHY-LSI.
USB 2.0 host/function module	VCC_USB	Input	Power supply pin.
	VSS_USB	Input	Ground pin.
	USB0_DP, USB1_DP	I/O	Inputs or outputs USB transceiver D+ data.
	USB0_DM, USB1_DM	I/O	Inputs or outputs USB transceiver D- data.
	USB0_VBUS, USB1_VBUS	Input	Input pins for detection of connection and disconnection of the USB cable.
	USB0_EXICEN	Output	Output pin for control the low power of the OTG chip.
	USB0_VBUSEN	Output	Supply enable pin of VBUS (5 V) for the OTG chip.
	USB0_OVRCURA, USB0_OVRCURB,	Input	Input pin for detection of external over current.
	USB0_ID	Input	ID input pin of mini-AB connector at the OGT operation.
	USB0_DPUPE, USB1_DPUPE	Output	Pull-up control pins of the D+ signal at the function operation.
	USB0_DPRPD	Output	Pull-down control pins of the D+ signal at the host operation.
	USB0_DRPD	Output	Pull-down control pins of the D- signal at the host operation.
CAN module	CRX0 to CRX2	Input	Input pin.
	CTX0 to CTX2	Output	Output pin.

Table 1.4 Pin Functions (5/6)

Classifications	Pin Name	I/O	Description
Serial peripheral interface	RSPCKA, RSPCKB RSPCKC	I/O	Clock input/output pin.
	MOSIA, MOSIB, MOSIC	I/O	Inputs or outputs data output from the master.
	MISOA, MISOB, MISOC	I/O	Inputs or outputs data output from the slave.
	SSLA0, SSLB0, SSLC0	I/O	Input or output pins slave selection
	SSLA1 to SSLA3 SSLB1 to SSLB3 SSLC1 to SSLC3	Output	Output pins slave selection
IEBus controller	IERXD	Input	Input pin for data reception.
	IETXD	Output	Output pin for data transmission.
Realtime clock	RTCOUT	Output	Output pin for 1-Hz clock.
	RTCIC0 to RTCIC2	Input	Time capture event input pin
12-bit A/D converter	AN000 to AN020	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0#	Input	Input pins for the external trigger signals that start the A/D conversion.
10-bit A/D converter	AN0 to AN7	Input	Input pins for the analog signals to be processed by the A/D converter.
	ANEX0	Output	Extended analog output pin
	ANEX1	Input	Extended analog input pin
	ADTRG#	Input	Input pins for the external trigger signals that start the A/D conversion.
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. Connect this pin to VCC if the 12-bit A/D converter is not to be used.
	AVSS0	Input	Analog ground pin for the 12-bit A/D converter. Connect this pin to VSS if the 12-bit A/D converter is not to be used.
	VREFH0	Input	Analog reference voltage supply pin for the 12-bit A/D converter. Connect this pin to VCC if the 12-bit A/D converter is not to be used.
Analog power supply	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter. Connect this pin to VSS if the 12-bit A/D converter is not to be used.
	VREFH	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This is used as the analog power supply for the respective modules. Connect this pin to VCC if neither the 10-bit A/D converter nor the D/A converter is in use.
	VREFL	Input	Reference ground pin for the 10-bit A/D converter and D/A converter. This is used as the analog ground for the respective modules. Set this pin to the same potential as the VSS pin.

Table 1.4 Pin Functions (6/6)

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/output pins)
	P40 to P47	I/O	8-bit input/output pins.
	P50 to P57	I/O	8-bit input/output pins.
	P60 to P67	I/O	8-bit input/output pins.
	P70 to P77	I/O	8-bit input/output pins.
	P80 to P87	I/O	8-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.

1.5 Pin Assignments

Figure 1.5 to Figure 1.8 show the pins assignments. Table 1.5 to Table 1.9 show the list of pins and pin functions. Power pins and I/O ports are shown in the pin assignment diagrams.

	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	P84	9
8	P94	PD1	PD2	VSS								P53	VCC_ USB	USB1_ DP	USB1_ DM	8
7	VSS	P92	PD0	P95								P54	P55	VSS_ USB	USB0_ DP	7
6	VCC	P91	P90	P93								P56	P57	VCC_ USB	USB0_ DM	6
5	P46	P47	P45	P44	NC							P13	P12	P10	P11	5
4	P42	P41	P43	P00	VSS	BSCANP	PF4	P35	PF3	PF1	P25	P86	P15	P14	P85	4
3	VREFL0	P40	VREFH0	P03	PF5	PJ3	MD/ FINED	RES#	P34	PF2	PF0	P24	P22	P87	P16	3
2	AVCC0	P07	VREFH	P02	EMLE	VCL	XCOUT	VSS	VCC	P32	P30	P26	P23	P17	P20	2
1	AVSS0	P05	VREFL	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	

RX63N Group
RX631 Group
PTLG0177KA-A
(177-pin TFLGA)
(Top perspective view)

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			RX63N Group RX631 Group PTBG0176GA-A (176-pin LFBGA) (Top perspective view)					P50	P51	P52	P84	9
8	P94	PD1	PD2	VSS								P53	VCC_ USB	USB1_ DP	USB1_ DM	8
7	VSS	P92	PD0	P95								P54	P55	VSS_ USB	USB0_ DP	7
6	VCC	P91	P90	P93								P56	P57	VCC_ USB	USB0_ DM	6
5	P46	P47	P45	P44								P13	P12	P10	P11	5
4	P42	P41	P43	P00	VSS	BSCANP	PF4	P35	PF3	PF1	P25	P86	P15	P14	P85	4
3	VREFL0	P40	VREFH0	P03	PF5	PJ3	MD/ FINED	RES#	P34	PF2	PF0	P24	P22	P87	P16	3
2	AVCC0	P07	VREFH	P02	EMLE	VCL	XCOUT	VSS	VCC	P32	P30	P26	P23	P17	P20	2
1	AVSS0	P05	VREFL	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	

Figure 1.4 Pin Assignment (176-Pin LFBGA)

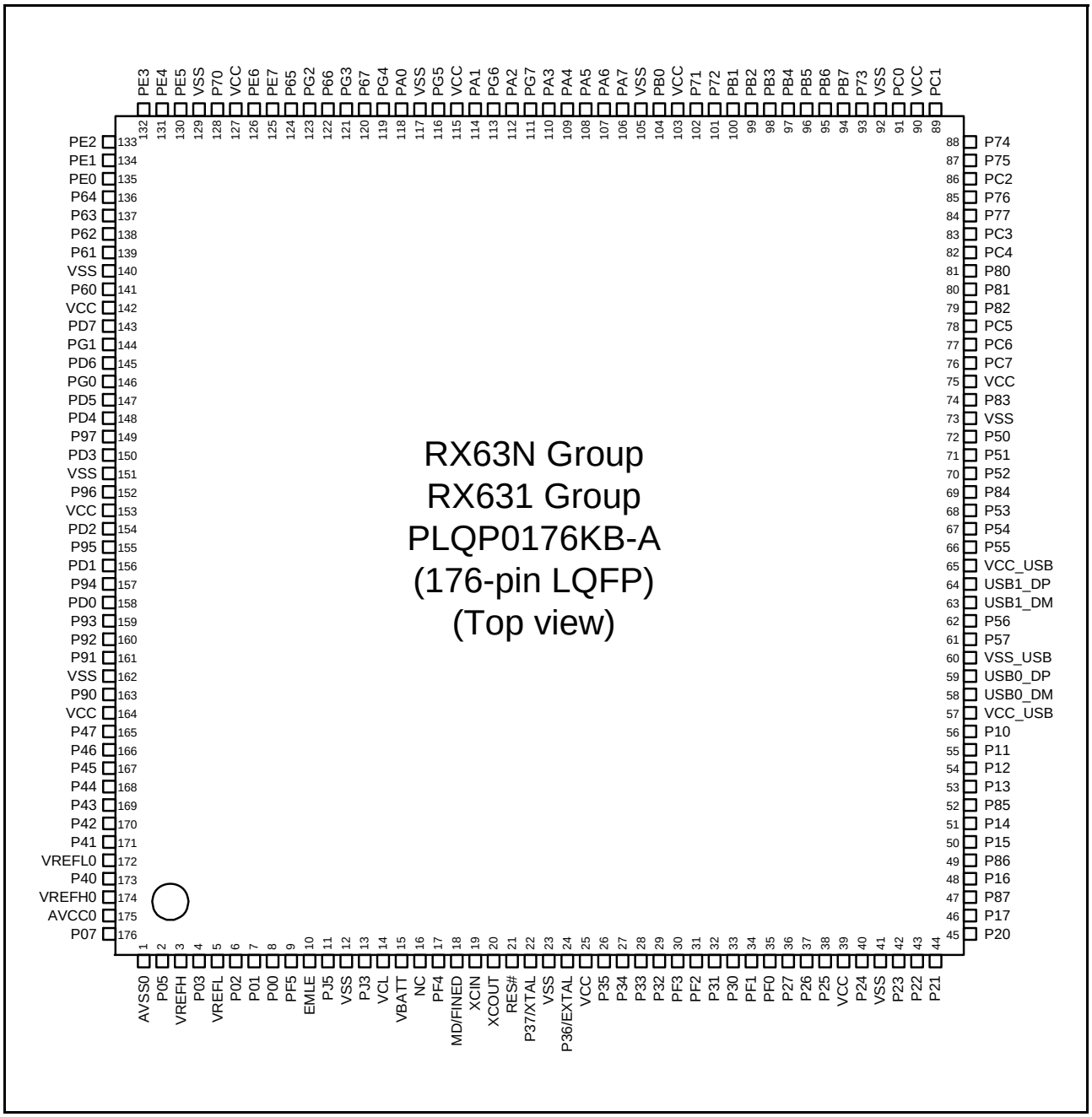
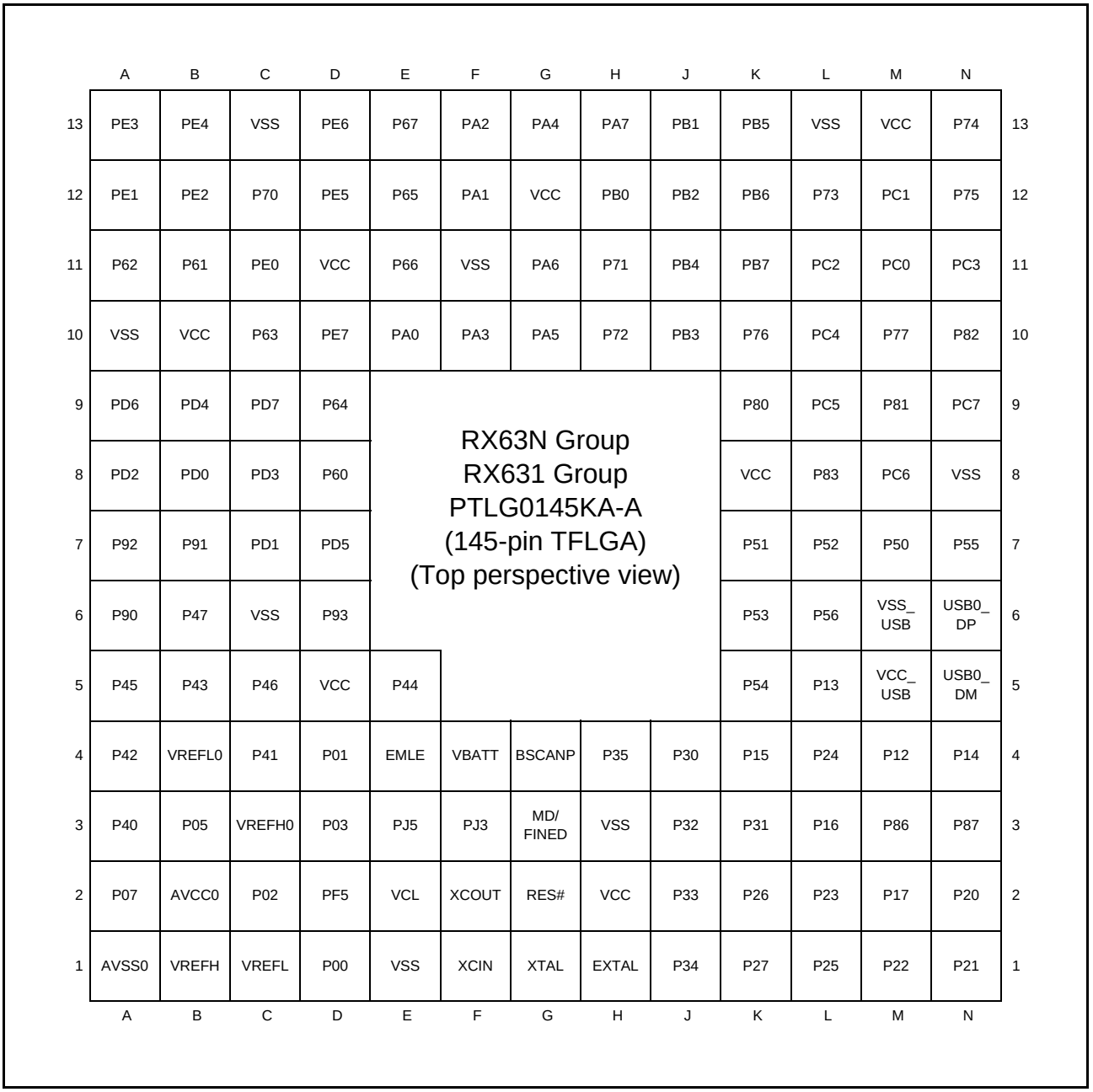


Figure 1.5 Pin Assignment (176-Pin LQFP)



	A	B	C	D	E	F	G	H	J	K	L	M	N	
13	PE3	PE4	VSS	PE6	P67	PA2	PA4	PA7	PB1	PB5	VSS	VCC	P74	13
12	PE1	PE2	P70	PE5	P65	PA1	VCC	PB0	PB2	PB6	P73	PC1	P75	12
11	P62	P61	PE0	VCC	P66	VSS	PA6	P71	PB4	PB7	PC2	PC0	PC3	11
10	VSS	VCC	P63	PE7	PA0	PA3	PA5	P72	PB3	P76	PC4	P77	P82	10
9	PD6	PD4	PD7	P64						P80	PC5	P81	PC7	9
8	PD2	PD0	PD3	P60						VCC	P83	PC6	VSS	8
7	P92	P91	PD1	PD5						P51	P52	P50	P55	7
6	P90	P47	VSS	P93						P53	P56	VSS_ USB	USB0_ DP	6
5	P45	P43	P46	VCC	P44					P54	P13	VCC_ USB	USB0_ DM	5
4	P42	VREFL0	P41	P01	EMLE	VBATT	BSCANP	P35	P30	P15	P24	P12	P14	4
3	P40	P05	VREFH0	P03	PJ5	PJ3	MD/ FINED	VSS	P32	P31	P16	P86	P87	3
2	P07	AVCC0	P02	PF5	VCL	XCOUT	RES#	VCC	P33	P26	P23	P17	P20	2
1	AVSS0	VREFH	VREFL	P00	VSS	XCIN	XTAL	EXTAL	P34	P27	P25	P22	P21	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	

Figure 1.6 Pin Assignment (145-Pin TFLGA)

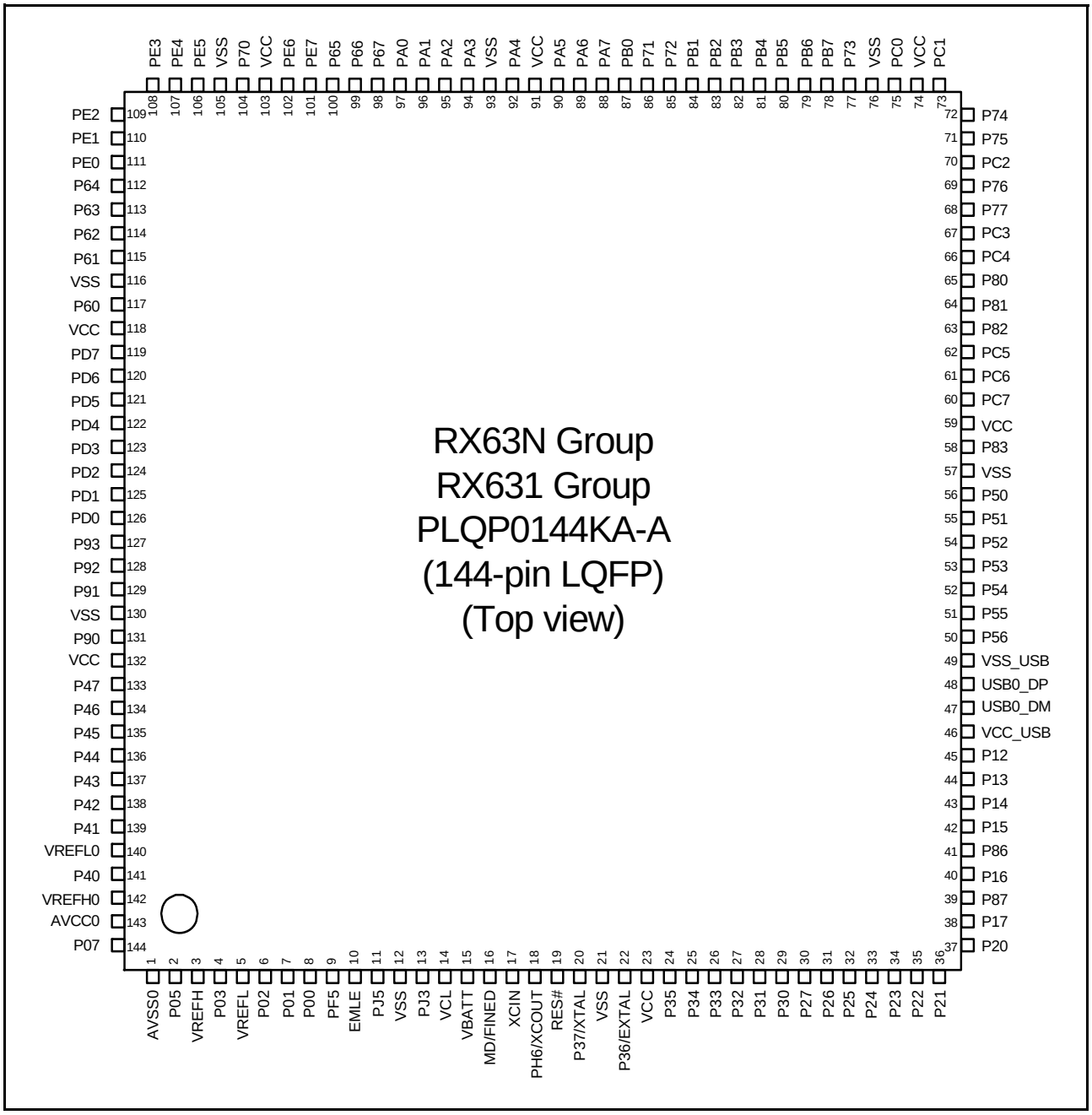


Figure 1.7 Pin Assignment (144-Pin LQFP)

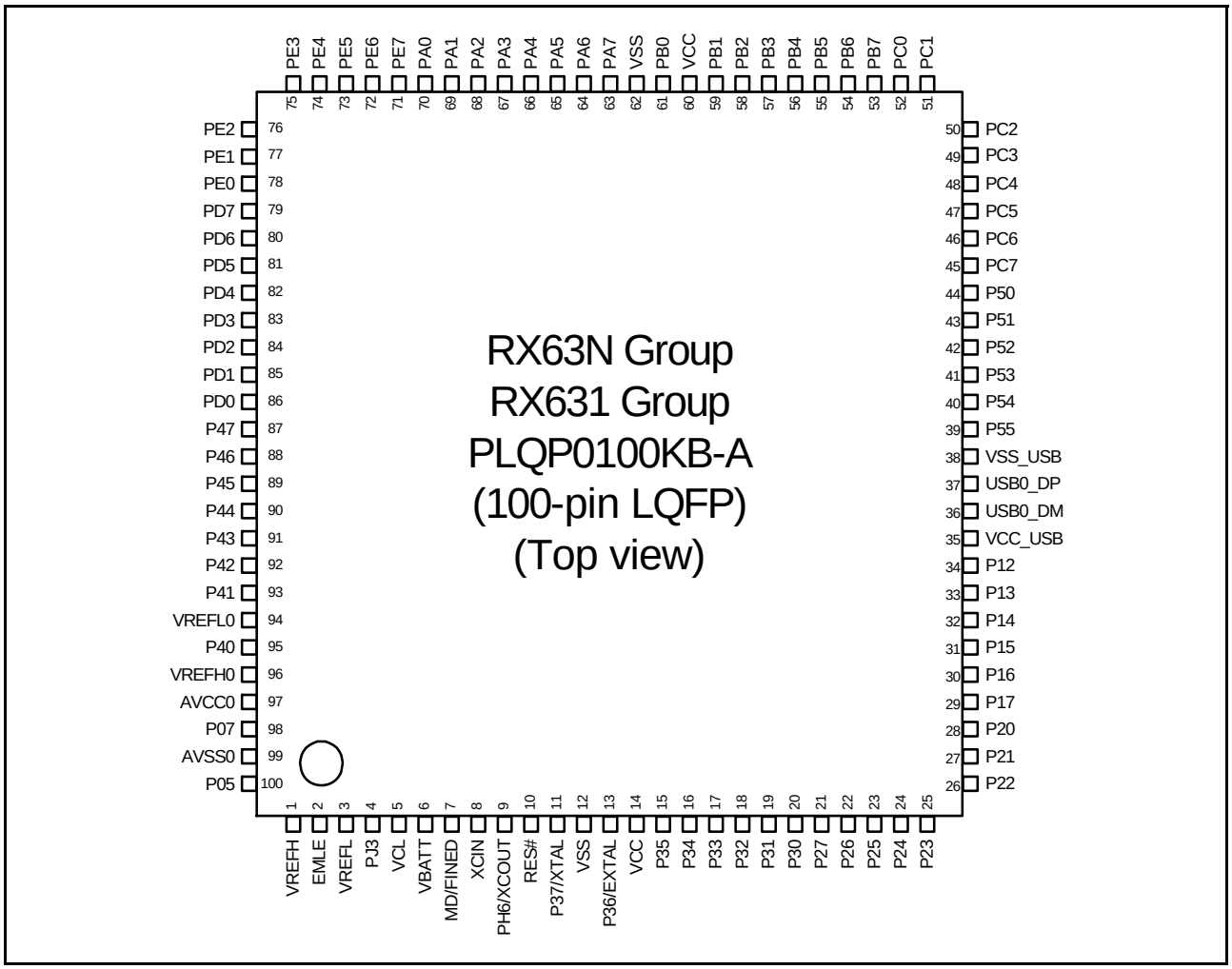


Figure 1.8 Pin Assignment (100-Pin LQFP)

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (1/5)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
A1	AVSS0						
A2	AVCC0						
A3	VREFL0						
A4		P42				IRQ10-DS	AN002
A5		P46				IRQ14-DS	AN006
A6	VCC						
A7	VSS						
A8		P94	A20/D20				
A9	VCC						
A10		P97	A23/D23				
A11		PD6	D6[A6/D6]	MTIC5V/POE1#	SSLC2	IRQ6	AN6
A12		P60	CS0#				
A13		P63	CS3#/CAS#				
A14		PE1	D9[A9/D9]	MTIOC4C/TIOCD9/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2/RSPCKB		ANEX1
A15		PE2	D10[A10/D10]	MTIOC4A/TIOCA9/PO23	RXD12/SMISO12/ SSCL12/RXDX12/SSLB3/ MOSIB	IRQ7-DS	AN0
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		SCK7		AN015
B7		P92	A18/D18		RXD7/SMISO7/SSCL7		AN016
B8		PD1	D1[A1/D1]	MTIOC4B/TIOCB7/ TCLKG	MOSIC/CTX0	IRQ1	AN009
B9		P96	A22/D22				
B10		PD4	D4[A4/D4]	POE3#	SSLC0	IRQ4	AN012
B11		PG1	D25				
B12	VSS						
B13		P64	CS4#/WE#				
B14		PE0	D8[A8/D8]	TIOCC9	SCK12/SSLB1		ANEX0
B15		PE3	D11[A11/D11]	MTIOC4B/TIOCB9/PO26/ POE8#	ET_ERXD3/CTS12#/ RTS12#/SS12#/MISOB		AN1
C1	VREFL						
C2	VREFH						
C3	VREFH0						
C4		P43				IRQ11-DS	AN003
C5		P45				IRQ13-DS	AN005
C6		P90	A16/D16		TXD7/SMOSI7/SSDA7		AN014
C7		PD0	D0[A0/D0]	TIOCA7		IRQ0	AN008
C8		PD2	D2[A2/D2]	MTIOC4D/TIOCA8	MISOC/CRX0	IRQ2	AN010
C9		PD3	D3[A3/D3]	TIOCB8/TCLKH/POE8#	RSPCKC	IRQ3	AN011
C10		PG0	D24				
C11	VCC						
C12		P62	CS2#/RAS#				
C13		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ TIOCA10/PO28	ET_ERXD2/SSLB0		AN2

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (2/5)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
C14	VSS						
C15	SDCLK	P70					
D1		P01		TMCI0	RXD6/SMISO6/SSCL6	IRQ9	AN019
D2		P02		TMCI1	SCK6	IRQ10	AN020
D3		P03				IRQ11	DA0
D4		P00		TMRI0	TXD6/SMOSI6/SSDA6	IRQ8	AN018
D5		P44				IRQ12-DS	AN004
D6		P93	A19/D19		CTS7#/RTS7#/SS7#		AN017
D7		P95	A21/D21				
D8	VSS						
D9		PD5	D5[A5/D5]	MTIC5W/POE2#	SSLC1	IRQ5	AN013
D10		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3	IRQ7	AN7
D11		P61	CS1#/SDCS#				
D12		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ TIOCB10	ET_RX_CLK/REF50CK/ RSPCKB	IRQ5	AN3
D13	VCC						
D14		PE7	D15[A15/D15]	TIOCB11	MISOB	IRQ7	AN5
D15		P65	CS5#/CKE				
E1		PJ5					
E2	EMLE						
E3		PF5				IRQ4	
E4	VSS						
E5*1	NC						
E12		PE6	D14[A14/D14]	TIOCA11	MOSIB	IRQ6	AN4
E13	TRDATA0	PG2	D26				
E14	TRDATA1	PG3	D27				
E15		P67	CS7#/DQM1		CRX2*3	IRQ15	
F1	VBATT						
F2	VCL						
F3		PJ3		MTIOC3C	CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#		
F4	BSCANP						
F12		P66	CS6#/DQM0		CTX2*3		
F13	TRSYNC#	PG4	D28				
F14		PA0	A0/BC0#/DQM2	MTIOC4A/TIOCA0/PO16	ET_TX_EN/ RMII_TXD_EN/SSLA1		
F15	VSS						
G1	XCIN						
G2	XCOUT						
G3	MD/FINED						
G4	TRST#	PF4					
G12	TRCLK	PG5	D29				
G13	TRDATA2	PG6	D30				
G14		PA1	A1/DQM3	MTIOC0B/MTCLKC/ TIOCB0/PO17	ET_WOL/SCK5/SSLA2	IRQ11	
G15	VCC						
H1	XTAL	P37					
H2	VSS						

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (3/5)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
H3	RES#						
H4		P35				NMI	
H12		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	ET_MDC/TXD5/SMOSI5/ SSDA5/SSLA0	IRQ5-DS	
H13		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	ET_MDIO/RXD5/SMISO5/ SSCL5	IRQ6-DS	
H14		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
H15	TRDATA3	PG7	D31				
J1	EXTAL	P36					
J2	VCC						
J3		P34		MTIOC0A/TMCI3/PO12/ POE2#	SCK6/SCK0/ USB0_DPRPD	IRQ4	
J4	TMS	PF3					
J12		PA5	A5	TIOCB1/PO21	ET_LINKSTA/RSPCKA		
J13	VSS						
J14		PA7	A7	TIOCB2/PO23	ET_WOL/MISOA		
J15		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	ET_EXOUT/CTS5#/ RTS5#/SS5#/MOSIA		
K1		P33		MTIOC0D/TIOCD0/ TMRI3/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0	IRQ3-DS	
K2		P32		MTIOC0C/TIOCC0/TMO3/ PO10/RTCOUT/RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/SSDA0/ CTX0/USB0_VBUSEN	IRQ2-DS	
K3	TDI	PF2			RXD1/SMISO1/SSCL1		
K4	TCK/FINEC	PF1			SCK1		
K12		PB2	A10	TIOCC3/TCLKC/PO26	ET_RX_CLK/REF50CK/ CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#		
K13		P71	CS1#		ET_MDIO		
K14	VCC						
K15		PB0	A8	MTIC5W/TIOCA3/PO24	ET_ERXD1/RMII_RXD1/ RXD4/RXD6/SMISO4/ SMISO6/SSCL4/SSCL6/ RSPCKA	IRQ12	
L1		P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0/USB0_DPUPE	IRQ1-DS	
L2		P30		MTIOC4B/TMRI3/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB/USB0_DRPD	IRQ0-DS	
L3	TDO	PF0			TXD1/SMOSI1/SSDA1		
L4		P25	CS5#/EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3/ USB0_DPRPD		ADTRG0#
L12		PB6	A14	MTIOC3D/TIOCA5/PO30	ET_ETXD1/RMII_TXD1/ RXD9/SMISO9/SSCL9		
L13		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	ET_RX_ER/RMII_RX_ER/ SCK4/SCK6		
L14		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	ET_ERXD0/RMII_RXD0/ TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/SSDA6	IRQ4-DS	
L15		P72	CS2#		ET_MDC		
M1		P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
M2		P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3# SMOSI1/SS3#/SSDA1/ MOSIB		

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (4/5)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
M3		P24	CS4#/EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/USB0_VBUSEN		
M4		P86		TIOCA0			
M5		P13		MTIOC0B/TIOCA5/TMO3/ PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
M6		P56	WR2#/BC2#/ EDACK1	MTIOC3C/TIOCA1			
M7		P54	ALE/EDACK0	MTIOC4B/TMCI1	ET_LINKSTA/CTS2#/ RTS2#/SS2#/CTX1		
M8		P53*2	BCLK				
M9		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
M10		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TIOCD6/TCLKF/TMRI2/ PO29	ET_ETXD2/SCK8/ RSPCKA		
M11		P81	EDACK0	MTIOC3D/PO27	ET_ETXD0/RMII_TXD0/ RXD10/SMISO10/SSCL10		
M12		P77	CS7#	PO23	ET_RX_ER/RMII_RX_ER/ TXD11/SMOSI11/SSDA11		
M13		PB7	A15	MTIOC3B/TIOCB5/PO31	ET_CRS/RMII_CRS_DV/ TXD9/SMOSI9/SSDA9		
M14		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	ET_ETXD0/RMII_TXD0/ SCK9		
M15		PB4	A12	TIOCA4/PO28	ET_TX_EN/ RMII_TXD_EN/CTS9#/ RTS9#/SS9#		
N1	VCC						
N2		P23	EDACK0	MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3/ USB0_DPUPE		
N3		P22	EDREQ0	MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0/USB0_DRPD		
N4		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS/ USB1_DPUPE	IRQ5	
N5		P12		MTIC5U/TMCI1	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
N6		P57	WAIT#/WR3#/ BC3#/EDREQ1				
N7		P55	WAIT#/ EDREQ0	MTIOC4D/TMO3	ET_EXOUT/CRX1	IRQ10	
N8	VCC_USB						
N9		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
N10		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TIOCB6/TMO2/PO31	ET_COL/TXD8/SMOSI8/ SSDA8/MISOA	IRQ14	
N11		P82	EDREQ1	MTIOC4A/PO28	ET_ETXD1/RMII_TXD1/ TXD10/SMOSI10/SSDA10		
N12		PC3	A19	MTIOC4D/TCLKB/PO24	ET_TX_ER/TXD5/ SMOSI5/SSDA5/IETXD		
N13		PC0	A16	MTIOC3C/TCLKC/PO17	ET_ERXD3/CTS5#/ RTS5#/SS5#/SSLA1/ SCL3	IRQ14	
N14		P73	CS3#	PO16	ET_WOL		
N15	VSS						
P1	VSS						

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

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
P2		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD/USB1_VBUS	IRQ7	ADTRG#
P3		P87		TIOCA2			
P4		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE/ USB0_OVRCURA	IRQ4	
P5		P10		MTIC5W/TMRI3		IRQ0	
P6	VCC_USB						
P7	VSS_USB						
P8					USB1_DP		
P9		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
P10		P83	EDACK1	MTIOC4C	ET_CRS/RMII_CRS_DV/ CTS10#/RTS10#/SS10#		
P11		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TIOCA6/TMCI2/PO30	ET_ETXD3/RXD8/ SMISO8/SSCL8/MOSIA	IRQ13	
P12		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TIOCC6/TCLKE/TMCI1/ PO25/POE0#	ET_TX_CLK/SCK5/ CTS8#/RTS8#/SS8#/ SSLA0		
P13		PC2	A18	MTIOC4B/TCLKA/PO21	ET_RX_DV/RXD5/ SMISO5/SSCL5/SSLA3/ IERXD		
P14		P75	CS5#	PO20	ET_ERXD0/RMII_RXD0/ SCK11		
P15	VCC						
R1		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0/ SCL1/USB0_EXICEN	IRQ9	
R2		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0/ SDA1/USB0_ID	IRQ8	
R3		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB	IRQ6	ADTRG0#
R4		P85					
R5		P11		MTIC5V/TMCI3	SCK2	IRQ1	
R6					USB0_DM		
R7					USB0_DP		
R8					USB1_DM		
R9		P84					
R10	VSS						
R11	VCC						
R12		P80	EDREQ0	MTIOC3B/PO26	ET_TX_EN/ RMII_TXD_EN/SCK10		
R13		P76	CS6#	PO22	ET_RX_CLK/REF50CK/ RXD11/SMISO11/SSCL11		
R14		P74	CS4#	PO19	ET_ERXD1/RMII_RXD1/ CTS11#/RTS11#/SS11#		
R15		PC1	A17	MTIOC3A/TCLKD/PO18	ET_ERXD2/SCK5/SSLA2/ SDA3	IRQ12	

Note 1. 176-pin LFBGA does not have 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.
Note 3. Enabled only for the on-chip ROM capacity: 2 Mbytes/1.5 Mbytes

Table 1.6 List of Pin and Pin Functions (176-Pin LQFP) (1/5)

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
1	AVSS0						
2		P05				IRQ13	DA1
3	VREFH						
4		P03				IRQ11	DA0
5	VREFL						
6		P02		TMCI1	SCK6	IRQ10	AN020
7		P01		TMCI0	RXD6/SMISO6/SSCL6	IRQ9	AN019
8		P00		TMRI0	TXD6/SMOSI6/SSDA6	IRQ8	AN018
9		PF5				IRQ4	
10	EMLE						
11		PJ5					
12	VSS						
13		PJ3		MTIOC3C	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		P35				NMI	
27		P34		MTIOC0A/TMCI3/PO12/ POE2#	SCK6/SCK0/ USB0_DPRPD	IRQ4	
28		P33		MTIOC0D/TIOCD0/ TMRI3/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0	IRQ3-DS	
29		P32		MTIOC0C/TIOCC0/TMO3/ PO10/RTCOUT/RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/SSDA0/ CTX0/USB0_VBUSEN	IRQ2-DS	
30	TMS	PF3					
31	TDI	PF2			RXD1/SMISO1/SSCL1		
32		P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0/USB0_DPUPE	IRQ1-DS	
33		P30		MTIOC4B/TMRI3/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB/USB0_DRPD	IRQ0-DS	
34	TCK/FINEC	PF1			SCK1		
35	TDO	PF0			TXD1/SMOSI1/SSDA1		
36		P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
37		P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3# SMOSI1/SS3#/SSDA1/ MOSIB		
38		P25	CS5#/EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3/ USB0_DPRPD		ADTRG0#
39	VCC						
40		P24	CS4#/EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/USB0_VBUSEN		

Table 1.6 List of Pin and Pin Functions (176-Pin LQFP) (2/5)

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
41	VSS						
42		P23	EDACK0	MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3/ USB0_DPUPE		
43		P22	EDREQ0	MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0/USB0_DRPD		
44		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0/ SCL1/USB0_EXICEN	IRQ9	
45		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0/ SDA1/USB0_ID	IRQ8	
46		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD/USB1_VBUS	IRQ7	ADTRG#
47		P87		TIOCA2			
48		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB	IRQ6	ADTRG0#
49		P86		TIOCA0			
50		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS/ USB1_DPUPE	IRQ5	
51		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE/ USB0_OVRCURA	IRQ4	
52		P85					
53		P13		MTIOC0B/TIOCA5/TMO3/ PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
54		P12		MTIC5U/TMCI1	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
55		P11		MTIC5V/TMCI3	SCK2	IRQ1	
56		P10		MTIC5W/TMRI3		IRQ0	
57	VCC_USB						
58					USB0_DM		
59					USB0_DP		
60	VSS_USB						
61		P57	WAIT#/WR3#/ BC3#/EDREQ1				
62		P56	WR2#/BC2#/ EDACK1	MTIOC3C/TIOCA1			
63					USB1_DM		
64					USB1_DP		
65	VCC_USB						
66		P55	WAIT#/ EDREQ0	MTIOC4D/TMO3	ET_EXOUT/CRX1	IRQ10	
67		P54	ALE/EDACK0	MTIOC4B/TMCI1	ET_LINKSTA/CTS2#/ RTS2#/SS2#/CTX1		
68		P53*1	BCLK				
69		P84					
70		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
71		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
72		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		

Table 1.6 List of Pin and Pin Functions (176-Pin LQFP) (3/5)

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
73	VSS						
74		P83	EDACK1	MTIOC4C	ET_CRS/RMII_CRS_DV/ CTS10#/RTS10#/SS10#		
75	VCC						
76		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TIOCB6/TMO2/PO31	ET_COL/TXD8/SMOSI8/ SSDA8/MISOA	IRQ14	
77		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TIOCA6/TMCI2/PO30	ET_ETXD3/RXD8/ SMISO8/SSCL8/MOSIA	IRQ13	
78		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TIOCD6/TCLKF/TMRI2/ PO29	ET_ETXD2/SCK8/ RSPCKA		
79		P82	EDREQ1	MTIOC4A/PO28	ET_ETXD1/RMII_TXD1/ TXD10/SMOSI10/SSDA10		
80		P81	EDACK0	MTIOC3D/PO27	ET_ETXD0/RMII_TXD0/ RXD10/SMISO10/SSCL10		
81		P80	EDREQ0	MTIOC3B/PO26	ET_TX_EN/ RMII_TXD_EN/SCK10		
82		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TIOCC6/TCLKE/TMCI1/ PO25/POE0#	ET_TX_CLK/SCK5/ CTS8#/RTS8#/SS8#/ SSLA0		
83		PC3	A19	MTIOC4D/TCLKB/PO24	ET_TX_ER/TXD5/ SMOSI5/SSDA5/IETXD		
84		P77	CS7#	PO23	ET_RX_ER/RMII_RX_ER/ TXD11/SMOSI11/SSDA11		
85		P76	CS6#	PO22	ET_RX_CLK/REF50CK/ RXD11/SMISO11/SSCL11		
86		PC2	A18	MTIOC4B/TCLKA/PO21	ET_RX_DV/RXD5/ SMISO5/SSCL5/SSLA3/ IERXD		
87		P75	CS5#	PO20	ET_ERXD0/RMII_RXD0/ SCK11		
88		P74	CS4#	PO19	ET_ERXD1/RMII_RXD1/ CTS11#/RTS11#/SS11#		
89		PC1	A17	MTIOC3A/TCLKD/PO18	ET_ERXD2/SCK5/SSLA2/ SDA3	IRQ12	
90	VCC						
91		PC0	A16	MTIOC3C/TCLKC/PO17	ET_ERXD3/CTS5#/ RTS5#/SS5#/SSLA1/ SCL3	IRQ14	
92	VSS						
93		P73	CS3#	PO16	ET_WOL		
94		PB7	A15	MTIOC3B/TIOCB5/PO31	ET_CRS/RMII_CRS_DV/ TXD9/SMOSI9/SSDA9		
95		PB6	A14	MTIOC3D/TIOCA5/PO30	ET_ETXD1/RMII_TXD1/ RXD9/SMISO9/SSCL9		
96		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	ET_ETXD0/RMII_TXD0/ SCK9		
97		PB4	A12	TIOCA4/PO28	ET_TX_EN/ RMII_TXD_EN/CTS9#/ RTS9#/SS9#		
98		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	ET_RX_ER/RMII_RX_ER/ SCK4/SCK6		
99		PB2	A10	TIOCC3/TCLKC/PO26	ET_RX_CLK/REF50CK/ CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#		
100		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	ET_ERXD0/RMII_RXD0/ TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/SSDA6	IRQ4-DS	

Table 1.6 List of Pin and Pin Functions (176-Pin LQFP) (4/5)

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
101		P72	CS2#		ET_MDC		
102		P71	CS1#		ET_MDIO		
103	VCC						
104		PB0	A8	MTIC5W/TIOCA3/PO24	ET_ERXD1/RMII_RXD1/ RXD4/RXD6/SMISO4/ SMISO6/SSCL4/SSCL6/ RSPCKA	IRQ12	
105	VSS						
106		PA7	A7	TIOCB2/PO23	ET_WOL/MISOA		
107		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	ET_EXOUT/CTS5#/ RTS5#/SS5#/MOSIA		
108		PA5	A5	TIOCB1/PO21	ET_LINKSTA/RSPCKA		
109		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	ET_MDC/TXD5/SMOSI5/ SSDA5/SSLA0	IRQ5-DS	
110		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	ET_MDIO/RXD5/SMISO5/ SSCL5	IRQ6-DS	
111	TRDATA3	PG7	D31				
112		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
113	TRDATA2	PG6	D30				
114		PA1	A1/DQM3	MTIOC0B/MTCLKC/ TIOCB0/PO17	ET_WOL/SCK5/SSLA2	IRQ11	
115	VCC						
116	TRCLK	PG5	D29				
117	VSS						
118		PA0	A0/BC0#/DQM2	MTIOC4A/TIOCA0/PO16	ET_TX_EN/ RMII_TXD_EN/SSLA1		
119	TRSYNC#	PG4	D28				
120		P67	CS7#/DQM1		CRX2*2	IRQ15	
121	TRDATA1	PG3	D27				
122		P66	CS6#/DQM0		CTX2*2		
123	TRDATA0	PG2	D26				
124		P65	CS5#/CKE				
125		PE7	D15[A15/D15]	TIOCB11	MISOB	IRQ7	AN5
126		PE6	D14[A14/D14]	TIOCA11	MOSIB	IRQ6	AN4
127	VCC						
128	SDCLK	P70					
129	VSS						
130		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ TIOCB10	ET_RX_CLK/REF50CK/ RSPCKB	IRQ5	AN3
131		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ TIOCA10/PO28	ET_ERXD2/SSLB0		AN2
132		PE3	D11[A11/D11]	MTIOC4B/TIOCB9/PO26/ POE8#	ET_ERXD3/CTS12#/ RTS12#/SS12#/MISOB		AN1
133		PE2	D10[A10/D10]	MTIOC4A/TIOCA9/PO23	RXD12/SMISO12/ SSCL12/RXDX12/SSLB3/ MOSIB	IRQ7-DS	AN0
134		PE1	D9[A9/D9]	MTIOC4C/TIOCD9/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2/RSPCKB		ANEX1
135		PE0	D8[A8/D8]	TIOCC9	SCK12/SSLB1		ANEX0
136		P64	CS4#/WE#				
137		P63	CS3#/CAS#				

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

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
138		P62	CS2#/RAS#				
139		P61	CS1#/SDCS#				
140	VSS						
141		P60	CS0#				
142	VCC						
143		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3	IRQ7	AN7
144		PG1	D25				
145		PD6	D6[A6/D6]	MTIC5V/POE1#	SSLC2	IRQ6	AN6
146		PG0	D24				
147		PD5	D5[A5/D5]	MTIC5W/POE2#	SSLC1	IRQ5	AN013
148		PD4	D4[A4/D4]	POE3#	SSLC0	IRQ4	AN012
149		P97	A23/D23				
150		PD3	D3[A3/D3]	TIOCB8/TCLKH/POE8#	RSPCKC	IRQ3	AN011
151	VSS						
152		P96	A22/D22				
153	VCC						
154		PD2	D2[A2/D2]	MTIOC4D/TIOCA8	MISOC/CRX0	IRQ2	AN010
155		P95	A21/D21				
156		PD1	D1[A1/D1]	MTIOC4B/TIOCB7/ TCLKG	MOSIC/CTX0	IRQ1	AN009
157		P94	A20/D20				
158		PD0	D0[A0/D0]	TIOCA7		IRQ0	AN008
159		P93	A19/D19		CTS7#/RTS7#/SS7#		AN017
160		P92	A18/D18		RXD7/SMISO7/SSCL7		AN016
161		P91	A17/D17		SCK7		AN015
162	VSS						
163		P90	A16/D16		TXD7/SMOSI7/SSDA7		AN014
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						
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.

Note 2. Enabled only for the on-chip ROM capacity: 2 Mbytes/1.5 Mbytes

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

Pin No. 145-pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
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		AN014
A7		P92	A18		RXD7/SMISO7/SSCL7		AN016
A8		PD2	D2[A2/D2]	MTIOC4D/TIOCA8	MISOC/CRX0	IRQ2	AN010
A9		PD6	D6[A6/D6]	MTIC5V/POE1#	SSLC2	IRQ6	AN6
A10	VSS						
A11		P62	CS2#/RAS#				
A12		PE1	D9[A9/D9]	MTIOC4C/TIOCD9/ PO18	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/SSLB2/ RSPCKB		ANEX1
A13		PE3	D11[A11/D11]	MTIOC4B/TIOCB9/ PO26/POE8#	CTS12#/RTS12#/SS12#/ MISOB/ET_ERXD3		AN1
B1	VREFH						
B2	AVCC0						
B3		P05				IRQ13	DA1
B4	VREFL0						
B5		P43				IRQ11-DS	AN003
B6		P47				IRQ15-DS	AN007
B7		P91	A17		SCK7		AN015
B8		PD0	D0[A0/D0]	TIOCA7		IRQ0	AN008
B9		PD4	D4[A4/D4]	POE3#	SSLC0	IRQ4	AN012
B10	VCC						
B11		P61	CS1#/SDCS#				
B12		PE2	D10[A10/D10]	MTIOC4A/TIOCA9/ PO23	RXD12/SMISO12/SSCL12/ RXDX12/SSLB3/MOSIB	IRQ7-DS	AN0
B13		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ TIOCA10/PO28	SSLB0/ET_ERXD2		AN2
C1	VREFL						
C2		P02		TMCI1	SCK6	IRQ10	AN020
C3	VREFH0						
C4		P41				IRQ9-DS	AN001
C5		P46				IRQ14-DS	AN006
C6	VSS						
C7		PD1	D1[A1/D1]	MTIOC4B/TIOCB7/ TCLKG	MOSIC/CTX0	IRQ1	AN009
C8		PD3	D3[A3/D3]	TIOCB8/TCLKH/POE8#	RSPCKC	IRQ3	AN011
C9		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3	IRQ7	AN7
C10		P63	CS3#/CAS#				
C11		PE0	D8[A8/D8]	TIOCC9	SCK12/SSLB1		ANEX0
C12	SDCLK	P70					
C13	VSS						
D1		P00		TMRI0	TXD6/SMOSI6/SSDA6	IRQ8	AN018
D2		PF5				IRQ4	
D3		P03				IRQ11	DA0
D4		P01		TMCI0	RXD6/SMISO6/SSCL6	IRQ9	AN019

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

Pin No. 145-pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
D5	VCC						
D6		P93	A19		CTS7#/RTS7#/SS7#		AN017
D7		PD5	D5[A5/D5]	MTIC5W/POE2#	SSLC1	IRQ5	AN013
D8		P60	CS0#				
D9		P64	CS4#/WE#				
D10		PE7	D15[A15/D15]	TIOCB11	MISOB	IRQ7	AN5
D11	VCC						
D12		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ TIOCB10	RSPCKB/ET_RX_CLK/ REF50CK	IRQ5	AN3
D13		PE6	D14[A14/D14]	TIOCA11		IRQ6	AN4
E1	VSS						
E2	VCL						
E3		PJ5					
E4	EMLE						
E5		P44				IRQ12-DS	AN004
E10		PA0	A0/BC0#	MTIOC4A/TIOCA0/ PO16	SSLA1/ET_TX_EN/ RMII_TXD_EN		
E11		P66	CS6#/DQM0		CTX2*2		
E12		P65	CS5#/CKE				
E13		P67	CS7#/DQM1		CRX2*2	IRQ15	
F1	XCIN						
F2	XCOUT						
F3		PJ3		MTIOC3C	CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#		
F4	VBATT						
F10		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5/ ET_MDIO	IRQ6-DS	
F11	VSS						
F12		PA1	A1	MTIOC0B/MTCLKC/ TIOCB0/PO17	SCK5/SSLA2/ET_WOL	IRQ11	
F13		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
G1	XTAL	P37					
G2	RES#						
G3	MD/FINED						
G4	BSCANP						
G10		PA5	A5	TIOCB1/PO21	RSPCKA/ET_LINKSTA		
G11		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	CTS5#/RTS5#/SS5# MOSIA/ET_EXOUT		
G12	VCC						
G13		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/ SSLA0/ET_MDC	IRQ5-DS	
H1	EXTAL	P36					
H2	VCC						
H3	VSS						
H4		P35				NMI	
H10		P72	CS2#		ET_MDC		
H11		P71	CS1#		ET_MDIO		

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

Pin No. 145-pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
H12		PB0	A8	MTIC5W/TIOCA3/PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/SSCL6/ RSPCKA/T_ERXD1/ RMII_RXD1	IRQ12	
H13		PA7	A7	TIOCB2/PO23	MISOA/ET_WOL		
J1	TRST#	P34		MTIOC0A/TMCI3/PO12/ POE2#	SCK6/SCK0/ USB0_DPRPD	IRQ4	
J2		P33		MTIOC0D/TIOCD0/ TMRI3/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0	IRQ3-DS	
J3		P32		MTIOC0C/TIOCC0/ TMO3/PO10/RTCOUT/ RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/SSDA0/ CTX0/USB0_VBUSEN	IRQ2-DS	
J4	TDI	P30		MTIOC4B/TMRI3/ PO8.RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB/USB0_DRPD	IRQ0-DS	
J10		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	SCK4/SCK6/ET_RX_ER/ RMII_RX_ER		
J11		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#/ ET_TX_EN/RMII_TXD_EN		
J12		PB2	A10	TIOCC3/TCLKC/PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#/ ET_RX_CLK/REF50CK		
J13		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/SSDA6/ ET_ERXD0/RMII_RXD0	IRQ4-DS	
K1	TCK/FINEC	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
K2	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1/ MOSIB		
K3	TMS	P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0/USB0_DPUPE	IRQ1-DS	
K4		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS	IRQ5	
K5	TRDATA2	P54	ALE/EDACK0	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1/ET_LINKSTA		
K6		P53*1	BCLK				
K7		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
K8	VCC						
K9	TRDATA0	P80	EDREQ0	MTIOC3B/PO26	SCK10/ET_TX_EN/ RMII_TXD_EN		
K10		P76	CS6#	PO22	RXD11/SMISO11/SSCL11/ ET_RX_CLK/REF50CK		
K11		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/SMOSI9/SSDA9/ ET_CRS/RMII_CRS_DV		
K12		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/SMISO9/SSCL9/ ET_ETXD1/RMII_TXD1		
K13		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9/ET_ETXD0/ RMII_TXD0		
L1		P25	CS5#/EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3/ USB0_DPRPD		ADTRG0#
L2		P23	EDACK0	MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3/ USB0_DPUPE		

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

Pin No. 145-pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
L3		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB	IRQ6	ADTRG0#
L4		P24	CS4#/EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/USB0_VBUSEN		
L5		P13		MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
L6		P56	EDACK1	MTIOC3C/TIOCA1			
L7		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
L8	TRCLK	P83	EDACK1	MTIOC4C	CTS10#/RTS10#/SS10#/ ET_CRS/RMII_CRS_DV		
L9		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TIOCD6/TCLKF/TMRI2/ PO29	SCK8/RSPCKA/ ET_ETXD2		
L10		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TIOCC6/TCLKE/TMCI1/ PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0/ET_TX_CLK		
L11		PC2	A18	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD/ET_RX_DV		
L12		P73	CS3#	PO16	ET_WOL		
L13	VSS						
M1		P22	EDREQ0	MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0/USB0_DRPD		
M2		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD	IRQ7	ADTRG#
M3		P86		TIOCA0			
M4		P12		TMCI1	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
M5	VCC_USB						
M6	VSS_USB						
M7		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
M8		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TIOCA6/TMCI2/PO30	RXD8/SMISO8/SSCL8/ MOSIA/ET_ETXD3	IRQ13	
M9	TRDATA1	P81	EDACK0	MTIOC3D/PO27	RXD10/SMISO10/SSCL10/ ET_ETXD0/RMII_TXD0		
M10		P77	CS7#	PO23	TXD11/SMOSI11/SSDA11/ ET_RX_ER/RMII_RX_ER		
M11		PC0	A16	MTIOC3C/TCLKC/PO17	CTS5#/RTS5#/SS5#/ SSLA1/SCL3/ET_ERXD3	IRQ14	
M12		PC1	A17	MTIOC3A/TCLKD/PO18	SCK5/SSLA2/SDA3/ ET_ERXD2	IRQ12	
M13	VCC						
N1		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0/ SCL1/USB0_EXICEN	IRQ9	
N2		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0/ SDA1/USB0_ID	IRQ8	
N3		P87		TIOCA2			
N4		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE/ USB0_OVRCURA	IRQ4	
N5					USB0_DM		
N6					USB0_DP		

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

Pin No. 145-pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
N7	TRDATA3	P55	WAIT#/ EDREQ0	MTIOC4D/TMO3	CRX1/ET_EXOUT	IRQ10	
N8	VSS						
N9		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TIOCB6/TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA/ET_COL	IRQ14	
N10	TRSYNC#	P82	EDREQ1	MTIOC4A/PO28	TXD10/SMOSI10/SSDA10/ ET_ETXD1/RMII_TXD1		
N11		PC3	A19	MTIOC4D/TCLKB/PO24	TXD5/SMOSI5/SSDA5/ IETXD/ET_TX_ER		
N12		P75	CS5#	PO20	SCK11/ET_ERXD0/ RMII_RXD0		
N13		P74	CS4#	PO19	CTS11#/RTS11#/SS11#/ ET_ERXD1/RMII_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.
Note 2. Enabled only for the on-chip ROM capacity: 2 Mbytes/1.5 Mbytes

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

Pin No. 144-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
1	AVSS0						
2		P05				IRQ13	DA1
3	VREFH						
4		P03				IRQ11	DA0
5	VREFL						
6		P02		TMCI1	SCK6	IRQ10	AN020
7		P01		TMCI0	RXD6/SMISO6/SSCL6	IRQ9	AN019
8		P00		TMRI0	TXD6/SMOSI6/SSDA6	IRQ8	AN018
9		PF5				IRQ4	
10	EMLE						
11		PJ5					
12	VSS						
13		PJ3		MTIOC3C	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/POE2#	SCK6/SCK0/ USB0_DPRPD	IRQ4	
26		P33		MTIOC0D/TIOCD0/ TMRI3/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0	IRQ3-DS	
27		P32		MTIOC0C/TIOCC0/ TMO3/PO10/RTCOUT/ RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN	IRQ2-DS	
28	TMS	P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0/USB0_DPUPE	IRQ1-DS	
29	TDI	P30		MTIOC4B/TMRI3/ PO8.RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB/USB0_DRPD	IRQ0-DS	
30	TCK/FINEC	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
31	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1/ MOSIB		
32		P25	CS5#/EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3/ USB0_DPRPD		ADTRG0#
33		P24	CS4#/EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/USB0_VBUSEN		
34		P23	EDACK0	MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3/ USB0_DPUPE		
35		P22	EDREQ0	MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0/USB0_DRPD		
36		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0/ SCL1/USB0_EXICEN	IRQ9	

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

Pin No. 144-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
37		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0/ SDA1/USB0_ID	IRQ8	
38		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/IETXD	IRQ7	ADTRG#
39		P87		TIOCA2			
40		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/ SSCL3/MOSIA/SCL2-DS/IERXD/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB	IRQ6	ADTRG0#
41		P86		TIOCA0			
42		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/ TMCI2/PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS	IRQ5	
43		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/ TMRI2/PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE/ USB0_OVRCURA	IRQ4	
44		P13		MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
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/ET_EXOUT	IRQ10	
52	TRDATA2	P54	ALE/EDACK0	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1/ET_LINKSTA		
53		P53*1	BCLK				
54		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
55		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
56		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
57	VSS						
58	TRCLK	P83	EDACK1	MTIOC4C	CTS10#/RTS10#/ SS10#/ET_CRS/ RMII_CRS_DV		
59	VCC						
60		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TIOCB6/TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA/ET_COL	IRQ14	
61		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TIOCA6/TMCI2/PO30	RXD8/SMISO8/SSCL8/ MOSIA/ET_ETXD3	IRQ13	
62		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TIOCD6/TCLKF/ TMRI2/PO29	SCK8/RSPCKA/ ET_ETXD2		
63	TRSYNC#	P82	EDREQ1	MTIOC4A/PO28	TXD10/SMOSI10/ SSDA10/ET_ETXD1/ RMII_TXD1		
64	TRDATA1	P81	EDACK0	MTIOC3D/PO27	RXD10/SMISO10/ SSCL10/ET_ETXD0/ RMII_TXD0		

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

Pin No. 144-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
65	TRDATA0	P80	EDREQ0	MTIOC3B/PO26	SCK10/ET_TX_EN/ RMII_TXD_EN		
66		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TIOCC6/TCLKE/ TMCI1/PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0/ ET_TX_CLK		
67		PC3	A19	MTIOC4D/TCLKB/ PO24	TXD5/SMOSI5/SSDA5/ IETXD/ET_TX_ER		
68		P77	CS7#	PO23	TXD11/SMOSI11/ SSDA11/ET_RX_ER/ RMII_RX_ER		
69		P76	CS6#	PO22	RXD11/SMISO11/ SSCL11/ET_RX_CLK/ REF50CK		
70		PC2	A18	MTIOC4B/TCLKA/ PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD/ ET_RX_DV		
71		P75	CS5#	PO20	SCK11/ET_ERXD0/ RMII_RXD0		
72		P74	CS4#	PO19	CTS11#/RTS11#/ SS11#/ET_ERXD1/ RMII_RXD1		
73		PC1	A17	MTIOC3A/TCLKD/ PO18	SCK5/SSLA2/SDA3/ ET_ERXD2	IRQ12	
74	VCC						
75		PC0	A16	MTIOC3C/TCLKC/ PO17	CTS5#/RTS5#/SS5#/ SSLA1/SCL3/ ET_ERXD3	IRQ14	
76	VSS						
77		P73	CS3#	PO16	ET_WOL		
78		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/SMOSI9/SSDA9/ ET_CRS/ RMII_CRS_DV		
79		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/SMISO9/SSCL9/ ET_ETXD1/RMII_TXD1		
80		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9/ET_ETXD0/ RMII_TXD0		
81		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#/ ET_TX_EN/ RMII_TXD_EN		
82		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	SCK4/SCK6/ ET_RX_ER/ RMII_RX_ER		
83		PB2	A10	TIOCC3/TCLKC/PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#/ ET_RX_CLK/REF50CK		
84		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/ SSDA6/ET_ERXD0/ RMII_RXD0	IRQ4-DS	
85		P72	CS2#		ET_MDC		
86		P71	CS1#		ET_MDIO		
87		PB0	A8	MTIC5W/TIOCA3/ PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/ SSCL6/RSPCKA/ T_ERXD1/RMII_RXD1	IRQ12	
88		PA7	A7	TIOCB2/PO23	MISOA/ET_WOL		
89		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	CTS5#/RTS5#/SS5# MOSIA/ET_EXOUT		
90		PA5	A5	TIOCB1/PO21	RSPCKA/ET_LINKSTA		

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

Pin No. 144-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
91	VCC						
92		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/ SSLA0/ET_MDC	IRQ5-DS	
93	VSS						
94		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5/ ET_MDIO	IRQ6-DS	
95		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
96		PA1	A1	MTIOC0B/MTCLKC/ TIOCB0/PO17	SCK5/SSLA2/ET_WOL	IRQ11	
97		PA0	A0/BC0#	MTIOC4A/TIOCA0/ PO16	SSLA1/ET_TX_EN/ RMII_TXD_EN		
98		P67	CS7#/DQM1		CRX2*2	IRQ15	
99		P66	CS6#/DQM0		CTX2*2		
100		P65	CS5#/CKE				
101		PE7	D15[A15/D15]	TIOCB11	MISOB	IRQ7	AN5
102		PE6	D14[A14/D14]	TIOCA11		IRQ6	AN4
103	VCC						
104	SDCLK	P70					
105	VSS						
106		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ TIOCB10	RSPCKB/ET_RX_CLK/ REF50CK	IRQ5	AN3
107		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ TIOCA10/PO28	SSLB0/ET_ERXD2		AN2
108		PE3	D11[A11/D11]	MTIOC4B/TIOCB9/ PO26/POE8#	CTS12#/RTS12#/ SS12#/MISOB/ ET_ERXD3		AN1
109		PE2	D10[A10/D10]	MTIOC4A/TIOCA9/ PO23	RXD12/SMISO12/ SSCL12/RXDX12/ SSLB3/MOSIB	IRQ7-DS	AN0
110		PE1	D9[A9/D9]	MTIOC4C/TIOCD9/ PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2/ RSPCKB		ANEX1
111		PE0	D8[A8/D8]	TIOCC9	SCK12/SSLB1		ANEX0
112		P64	CS4#/WE#				
113		P63	CS3#/CAS#				
114		P62	CS2#/RAS#				
115		P61	CS1#/SDCS#				
116	VSS						
117		P60	CS0#				
118	VCC						
119		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3	IRQ7	AN7
120		PD6	D6[A6/D6]	MTIC5V/POE1#	SSLC2	IRQ6	AN6
121		PD5	D5[A5/D5]	MTIC5W/POE2#	SSLC1	IRQ5	AN013
122		PD4	D4[A4/D4]	POE3#	SSLC0	IRQ4	AN012
123		PD3	D3[A3/D3]	TIOCB8/TCLKH/ POE8#	RSPCKC	IRQ3	AN011
124		PD2	D2[A2/D2]	MTIOC4D/TIOCA8	MISOC/CRX0	IRQ2	AN010
125		PD1	D1[A1/D1]	MTIOC4B/TIOCB7/ TCLKG	MOSIC/CTX0	IRQ1	AN009
126		PD0	D0[A0/D0]	TIOCA7		IRQ0	AN008
127		P93	A19		CTS7#/RTS7#/SS7#		AN017

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

Pin No. 144-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
128		P92	A18		RXD7/SMISO7/SSCL7		AN016
129		P91	A17		SCK7		AN015
130	VSS						
131		P90	A16		TXD7/SMOSI7/SSDA7		AN014
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.
Note 2. Enabled only for the on-chip ROM capacity: 2 Mbytes/1.5 Mbytes

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

Pin No. 100-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
1	VREFH						
2	EMLE						
3	VREFL						
4		PJ3		MTIOC3C	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		P35				NMI	
16	TRST#	P34		MTIOC0A/TMCI3/ PO12/POE2#	SCK6/SCK0/ USB0_DPRPD	IRQ4	
17		P33		MTIOC0D/TIOCD0/ TMRI3/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0*1	IRQ3-DS	
18		P32		MTIOC0C/TIOCC0/ TMO3/PO10/RTCOUT/ RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0*1/ USB0_VBUSEN	IRQ2-DS	
19	TMS	P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0/USB0_DPUPE	IRQ1-DS	
20	TDI	P30		MTIOC4B/TMRI3/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB/USB0_DRPD	IRQ0-DS	
21	TCK/FINEC	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
22	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1/ MOSIB		
23		P25	CS5#/EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3/ USB0_DPRPD		ADTRG0#
24		P24	CS4#/EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/USB0_VBUSEN		
25		P23	EDACK0	MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3/ USB0_DPUPE		
26		P22	EDREQ0	MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0/USB0_DRPD		
27		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0/ USB0_EXICEN	IRQ9	
28		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0/ USB0_ID	IRQ8	
29		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/IETXD	IRQ7	ADTRG#
30		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/ SSCL3/MOSIA/SCL2-DS/IERXD/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB	IRQ6	ADTRG0#

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

Pin No. 100-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
31		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS	IRQ5	
32		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE/ USB0_OVRCURA	IRQ4	
33		P13		MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
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/ET_EXOUT	IRQ10	
40		P54	ALE/EDACK0	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1/ET_LINKSTA		
41		P53*2	BCLK				
42		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
43		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
44		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
45		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA/ET_COL	IRQ14	
46		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TMCI2/PO30	RXD8/SMISO8/SSCL8/ MOSIA/ET_ETXD3	IRQ13	
47		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TMRI2/PO29	SCK8/RSPCKA/ ET_ETXD2		
48		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TMCI1/PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0/ ET_TX_CLK		
49		PC3	A19	MTIOC4D/TCLKB/ PO24	TXD5/SMOSI5/SSDA5/ IETXD/ET_TX_ER		
50		PC2	A18	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD/ ET_RX_DV		
51		PC1	A17	MTIOC3A/TCLKD/ PO18	SCK5/SSLA2/ ET_ERXD2	IRQ12	
52		PC0	A16	MTIOC3C/TCLKC/ PO17	CTS5#/RTS5#/SS5#/ SSLA1/ET_ERXD3	IRQ14	
53		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/SMOSI9/SSDA9/ ET_CRS/ RMII_CRS_DV		
54		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/SMISO9/SSCL9/ ET_ETXD1/RMII_TXD1		
55		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9/ET_ETXD0/ RMII_TXD0		
56		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#/ ET_TX_EN/ RMII_TXD_EN		
57		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	SCK6/ET_RX_ER/ RMII_RX_ER		
58		PB2	A10	TIOCC3/TCLKC/PO26	CTS6#/RTS6#/SS6#/ ET_RX_CLK/REF50CK		

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

Pin No. 100-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
59		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD6/SMOSI6/SSDA6/ ET_ERXD0/ RMII_RXD0	IRQ4-DS	
60	VCC						
61		PB0	A8	MTIC5W/TIOCA3/PO24	RXD6/SMISO6/SSCL6/ RSPCKA/ET_ERXD1/ RMII_RXD1	IRQ12	
62	VSS						
63		PA7	A7	TIOCB2/PO23	MISOA/ET_WOL		
64		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	CTS5#/RTS5#/SS5#/ MOSIA/ET_EXOUT		
65		PA5	A5	TIOCB1/PO21	RSPCKA/ET_LINKSTA		
66		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/ SSLA0/ET_MDC	IRQ5-DS	
67		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5/ ET_MDIO	IRQ6-DS	
68		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
69		PA1	A1	MTIOC0B/MTCLKC/ TIOCB0/PO17	SCK5/SSLA2/ET_WOL	IRQ11	
70		PA0	A0/BC0#	MTIOC4A/TIOCA0/ PO16	SSLA1/ET_TX_EN/ RMII_TXD_EN		
71		PE7	D15[A15/D15]		MISOB	IRQ7	AN5
72		PE6	D14[A14/D14]		MOSIB	IRQ6	AN4
73		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B	RSPCKB/ET_RX_CLK/ REF50CK	IRQ5	AN3
74		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ PO28	SSLB0/ET_ERXD2		AN2
75		PE3	D11[A11/D11]	MTIOC4B/PO26/POE8#	CTS12#/RTS12#/ SS12#/MISOB/ ET_ERXD3		AN1
76		PE2	D10[A10/D10]	MTIOC4A/PO23	RXD12/SMISO12/ SSCL12/RXDX12/ SSLB3/MOSIB	IRQ7-DS	AN0
77		PE1	D9[A9/D9]	MTIOC4C/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2/ RSPCKB		ANEX1
78		PE0	D8[A8/D8]		SCK12/SSLB1		ANEX0
79		PD7	D7[A7/D7]	MTIC5U/POE0#		IRQ7	AN7
80		PD6	D6[A6/D6]	MTIC5V/POE1#		IRQ6	AN6
81		PD5	D5[A5/D5]	MTIC5W/POE2#		IRQ5	AN013
82		PD4	D4[A4/D4]	POE3#		IRQ4	AN012
83		PD3	D3[A3/D3]	POE8#		IRQ3	AN011
84		PD2	D2[A2/D2]	MTIOC4D	CRX0*1	IRQ2	AN010
85		PD1	D1[A1/D1]	MTIOC4B	CTX0*1	IRQ1	AN009
86		PD0	D0[A0/D0]			IRQ0	AN008
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

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

Pin No. 100-pin LQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timers (MTU, TPU, TMR, PPG, RTC, POE)	Communications (ETHERC, SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD AD DA
94	VREFL0						
95		P40				IRQ8-DS	AN000
96	VREFH0						
97	AVCC0						
98		P07				IRQ15	ADTRG0#
99	AVSS0						
100		P05				IRQ13	DA1

Note 1. Enabled only for the on-chip ROM capacity of 768 Kbytes or more
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.

2. CPU

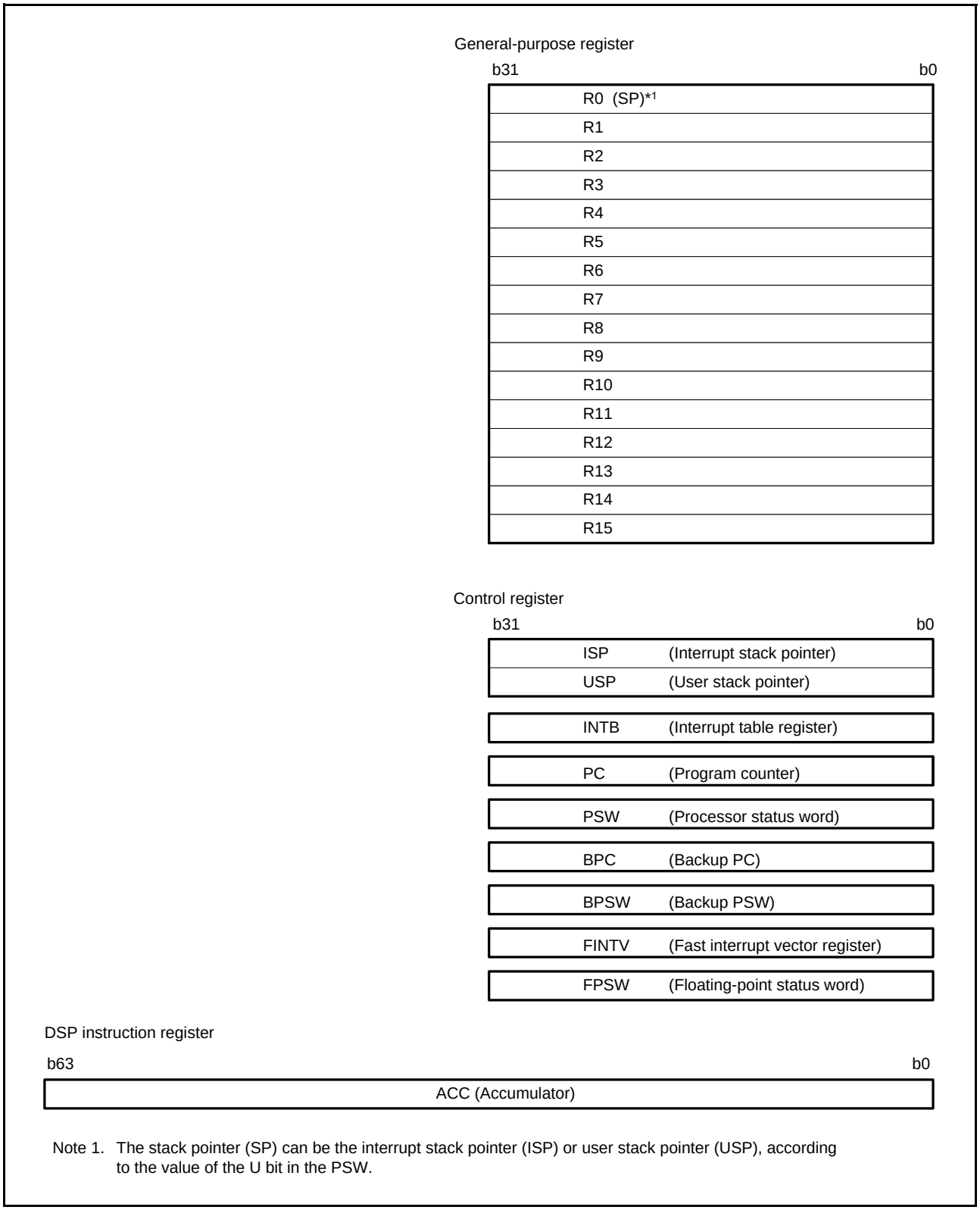
The RX63N/RX631 Group is an MCU with the high-speed, high-performance RX CPU as its core.
A variable-length instruction format has been adopted for the RX CPU. Allocating the more frequently used instructions to the shorter instruction lengths facilitates the development of efficient programs that take up less memory.
The CPU has 73 basic instructions and 8 floating-point operation instructions, and nine DSP instructions, for a total of 90 instructions. It has 10 addressing modes and caters to register–register operations, register–memory operations, immediate–register operations, immediate–memory operations, memory–memory transfer, and bitwise operations. High-speed operation was realized by achieving execution in a single cycle not only for register–register operations, but also for other types of multiple instructions. The CPU includes an internal multiplier and an internal divider for high-speed multiplication and division.
The RX CPU has a five-stage pipeline for processing instructions. The stages are instruction fetching, instruction decoding, execution, memory access, and write-back. In cases where pipeline processing is drawn-out by memory access, subsequent operations may in fact be executed earlier. By adopting “out-of-order completion” of this kind, the execution of instructions is controlled to optimize numbers of clock cycles.

2.1 Features

- High instruction execution rate: One instruction in one clock cycle
- Address space: 4-Gbyte linear
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Nine 32-bit registers
  Accumulator: One 64-bit register
- Basic instructions: 73 (arithmetic/logic instructions, data-transfer instructions, branch instructions, bit-manipulation instructions, string-manipulation instructions, and system-manipulation instructions)
  Relative branch instructions to suit branch distances
  Variable-length instruction format (lengths from one to eight bytes)
  Short formats for frequently used instructions
- Floating-point operation instructions: 8
- DSP instructions: 9
  Supports 16-bit × 16-bit multiplication and multiply-and-accumulate operations.
  Rounds the data in the accumulator.
- Addressing modes: 10
- Five-stage pipeline
  Adoption of out-of-order completion
- Processor modes
  A supervisor mode and a user mode are supported.
- Floating-point operation unit
  Supports single-precision (32-bit) floating point
  Supports data types and exceptions in conformance with the IEEE754 standard
- Data arrangement
  Selectable as little endian or big endian

2.2 Register Set of the CPU

The RX CPU has sixteen general-purpose registers, nine control registers, and one 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)

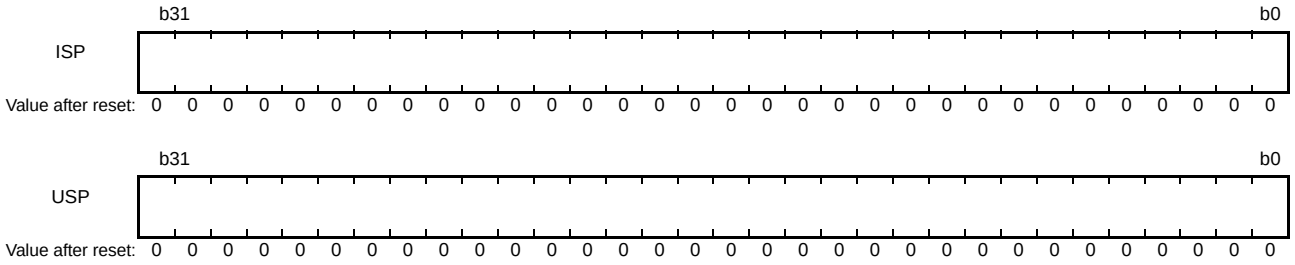
This CPU has sixteen general-purpose registers (R0 to R15). R1 to R15 can be used as data registers or address registers. R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) 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 nine control registers.

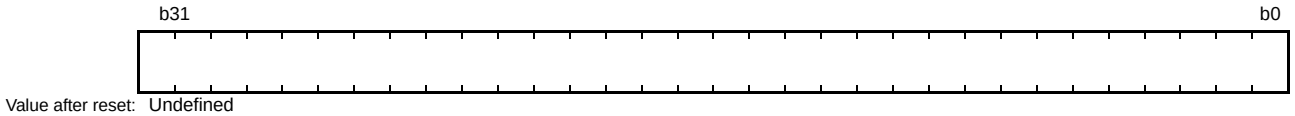
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Floating-point status word (FPSW)

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



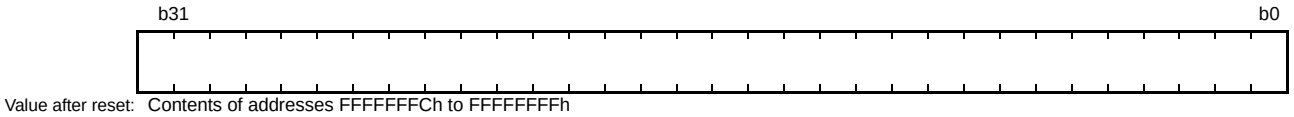
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).
Set the ISP or USP to a multiple of four, as this reduces the numbers of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.2 Interrupt Table Register (INTB)



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

2.2.2.3 Program Counter (PC)



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

2.2.2.4 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

Bit	Symbol	Bit Name	Description	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 indicates whether a carry, borrow, or shift-out has occurred as the result of an operation.

Z Flag (Zero Flag)

This flag indicates that the result of an operation was 0.

S Flag (Sign Flag)

This flag indicates that the result of an operation was negative.

O Flag (Overflow Flag)

This flag indicates that an overflow occurred during an operation.

I Bit (Interrupt Enable)

This bit enables interrupt requests. When an exception is accepted, the value of this bit becomes 0.

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.

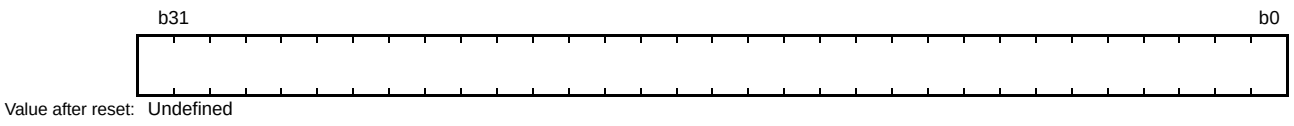
PM Bit (Processor Mode Select)

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)

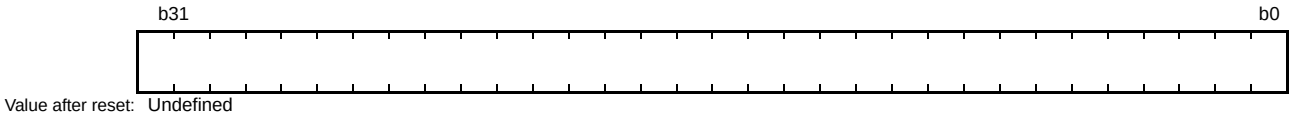
The IPL[3:0] bits specify the processor interrupt priority level as one of sixteen levels from zero to fifteen, wherein priority level zero is the lowest and priority level fifteen the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the 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.

2.2.2.5 Backup PC (BPC)



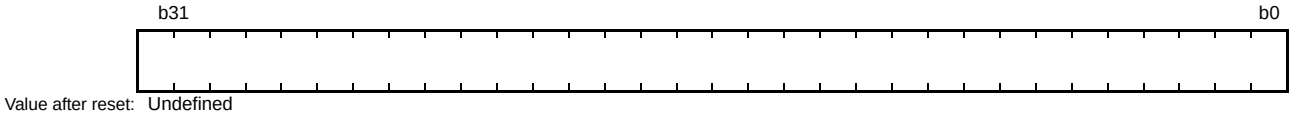
The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

2.2.2.6 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.

2.2.2.7 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.

2.2.2.8 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

Bit	Symbol	Bit Name	Description	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
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 cleared 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 cleared 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 cleared 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.

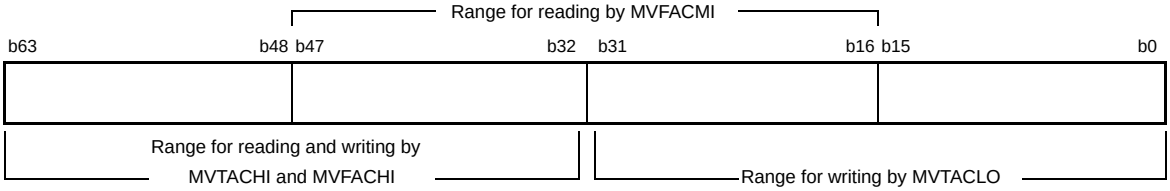
- When Ej is 1 (exception handling is enabled), the value of the flag remains.
- When the corresponding flag is set to 1, it remains 1 until it is cleared to 0 by software. (Accumulation flag)

FS Flag (Floating-Point Error Summary Flag)

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

2.2.3 Register Associated with DSP Instructions

2.2.3.1 Accumulator (ACC)



Value after reset: Undefined

The accumulator (ACC) is a 64-bit register used for DSP instructions. The accumulator is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, FMUL, MUL, and RMPA, in which case the prior value in the accumulator is modified by execution of the instruction.
Use the MVTACHI and MVTACLO instructions for writing to the accumulator. The MVTACHI and MVTACLO instructions write data to the higher-order 32 bits (bits 63 to 32) and the lower-order 32 bits (bits 31 to 0), respectively.
Use the MVFACHI and MVFACMI instructions for reading data from the accumulator. The MVFACHI and MVFACMI instructions read data from the higher-order 32 bits (bits 63 to 32) and the middle 32 bits (bits 47 to 16), respectively.

2.3 Processor Mode

The RX CPU supports two processor modes, supervisor and user. These processor modes enable the realization of a hierarchical CPU resource protection.
Each processor mode imposes a level on rights of access to the CPU resources and the instructions that can be executed. Supervisor mode carries greater rights than those of 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 2.2.2.4, 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)
- 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 RX CPU can handle four types of data: integer, floating-point, bit, and string.

2.4.1 Integer

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

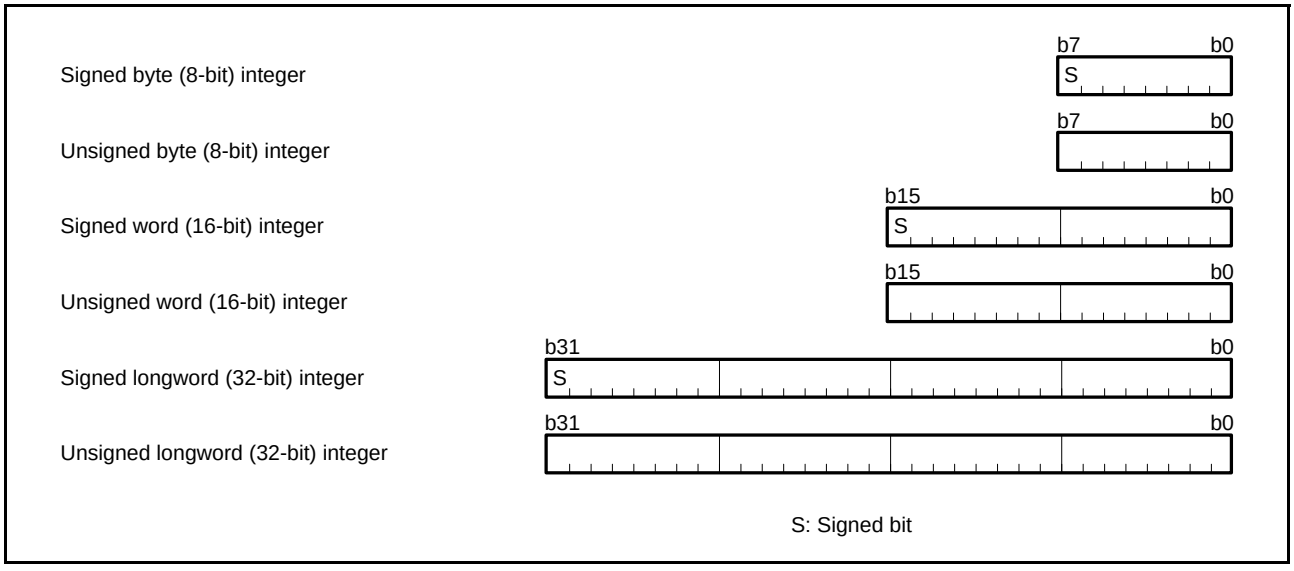


Figure 2.2 Integer

2.4.2 Floating-Point

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

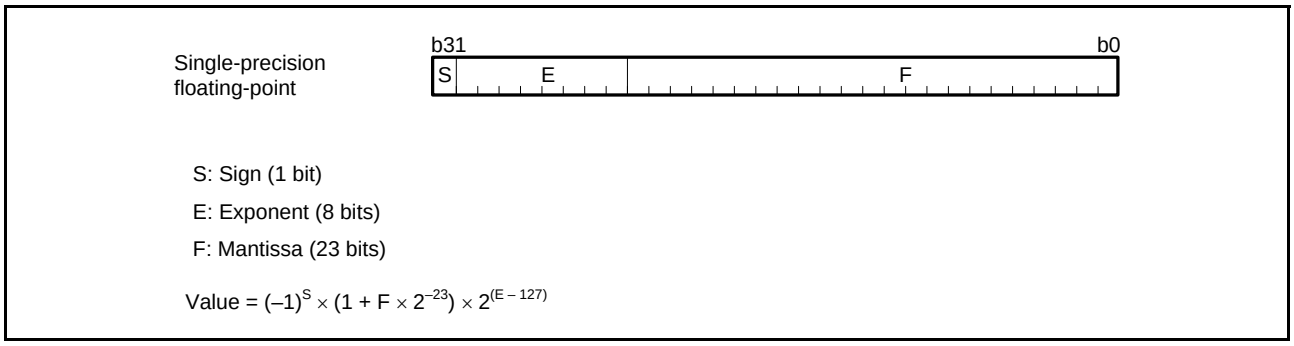


Figure 2.3 Floating-Point

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 Bit wise Operations

Five bit-manipulation instructions are provided for bitwise operations: BCLR, BMCnd, 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.

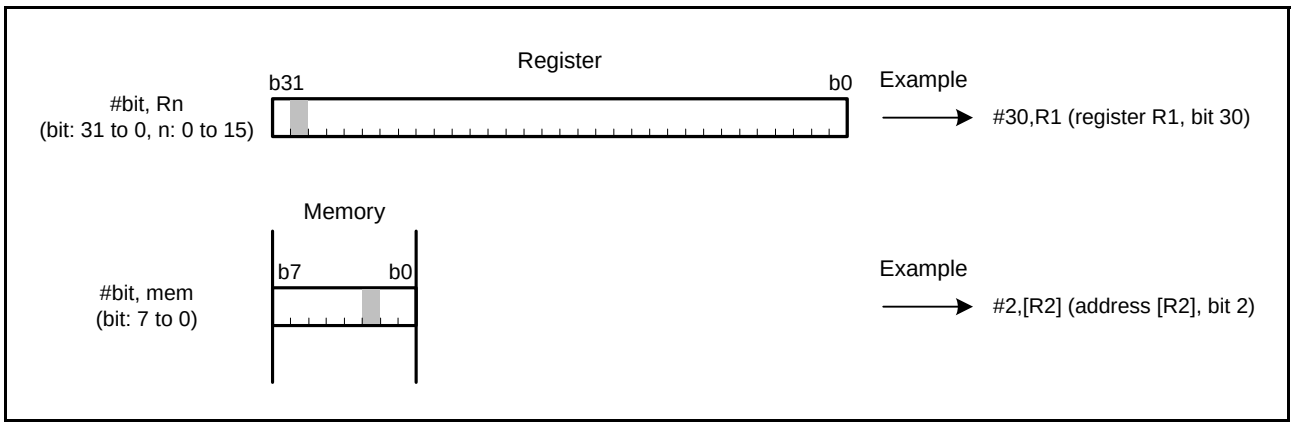


Figure 2.4 Bit

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.

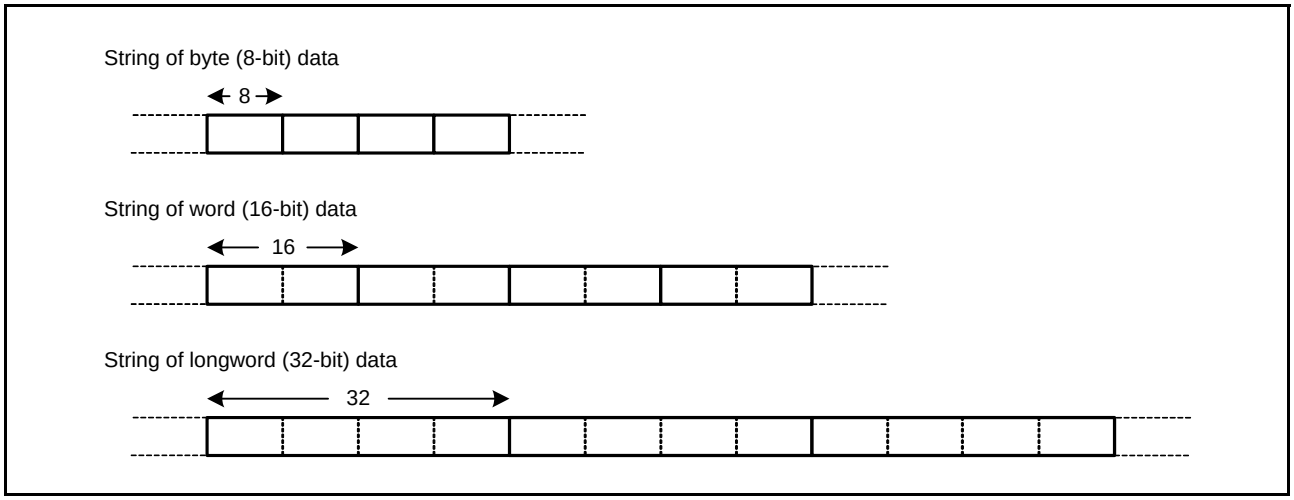


Figure 2.5 String

2.5 Endian

For the RX 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 RX63N/RX631 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 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 2.9 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.

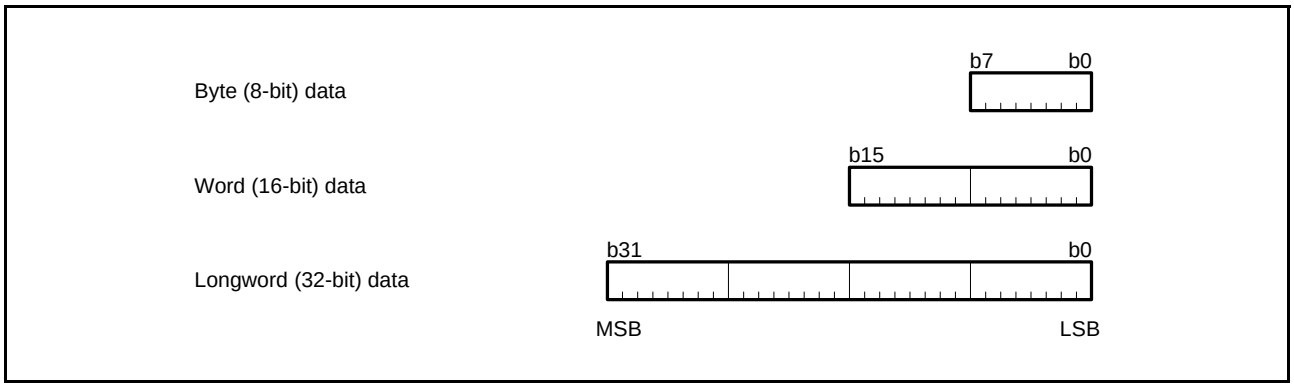


Figure 2.6 Data Arrangement in Registers

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.

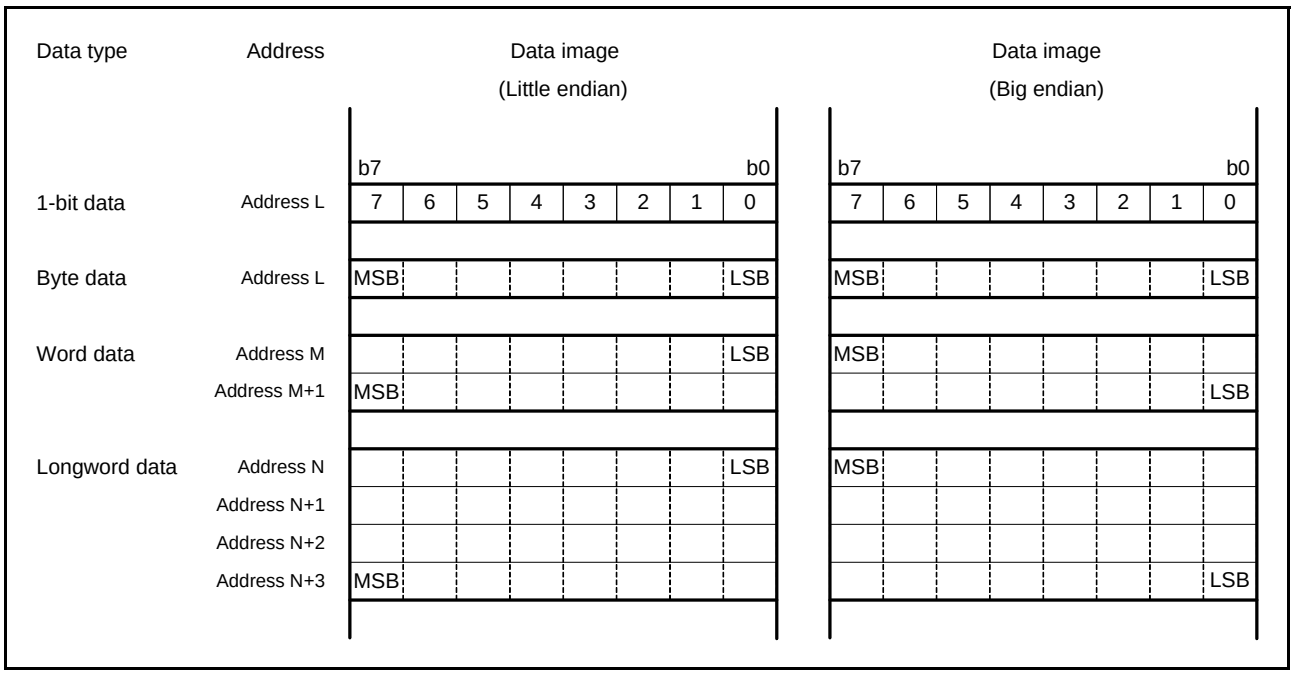


Figure 2.7 Data Arrangement in Memory

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: fixed and relocatable. Each vector in the vector table consists of four bytes and specifies the address where the corresponding exception handling routine starts.

2.6.1 Fixed Vector Table

The fixed vector table is allocated to a fixed address range. The individual vectors for the privileged instruction exception, undefined instruction exception, floating-point exception, non-maskable interrupt, and reset are allocated to addresses in the range from FFFFFF80h to FFFFFFFFh. Figure 2.8 shows the fixed vector table.

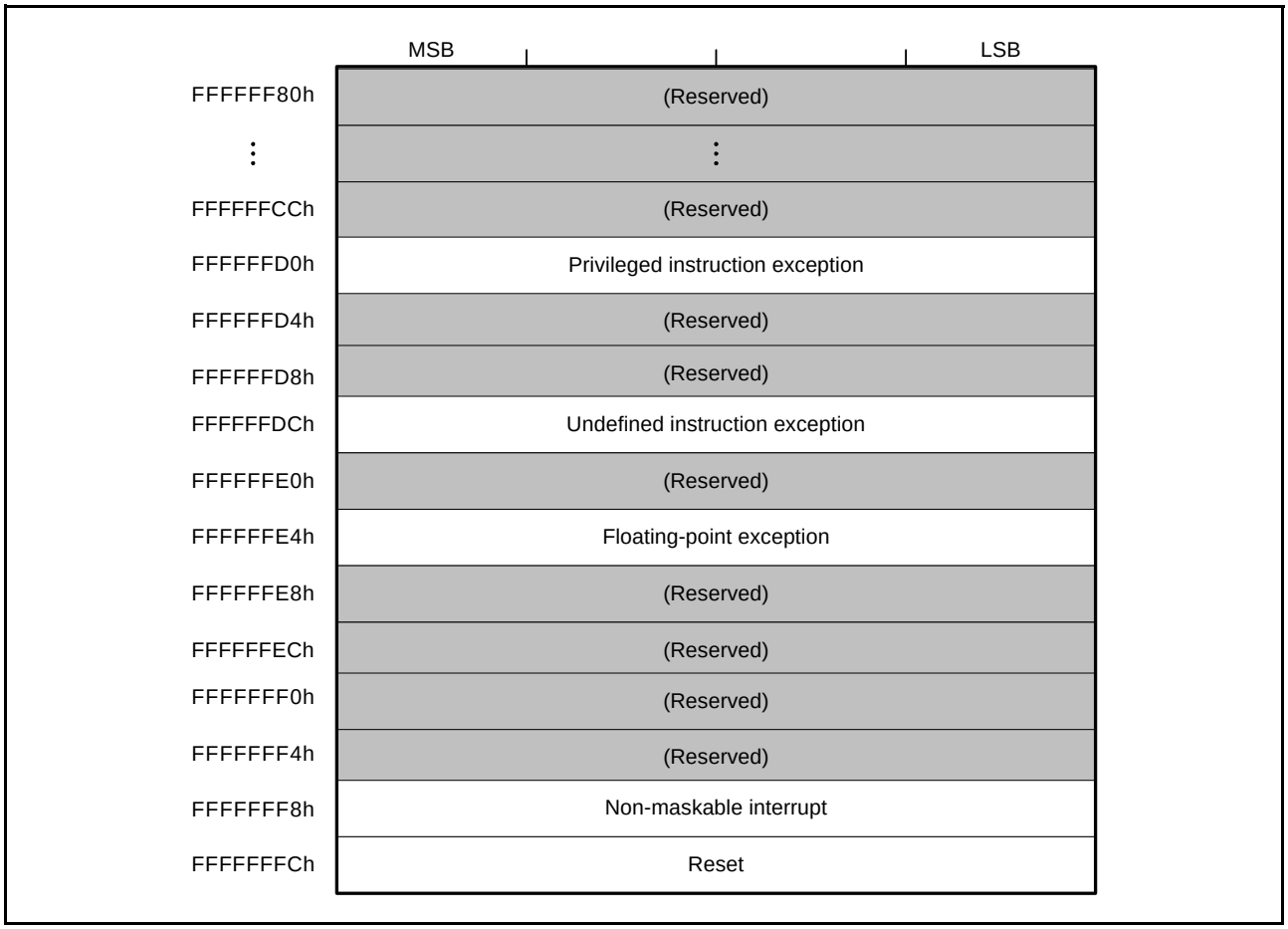


Figure 2.8 Fixed Vector Table

2.6.2 Relocatable Vector Table

The address where the relocatable 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 relocatable vector table.

Each vector in the relocatable 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.

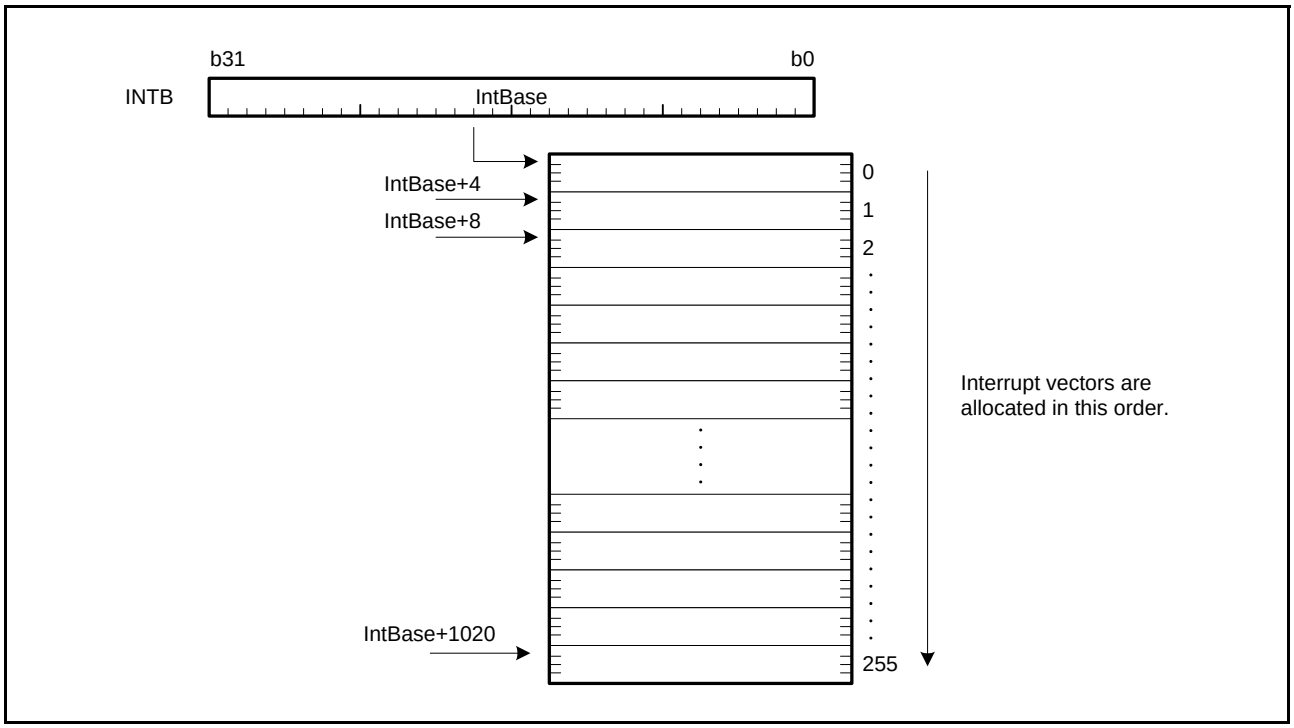


Figure 2.9 Relocatable Vector Table

2.7 Operation of Instructions

2.7.1 Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions

The RMPA instruction and the string-manipulation instructions except the SSTR instruction (that is, SCMPU, SMOVB, SMOVF, SMOVU, SUNTIL, and SWHILE instructions) may prefetch data from the memory to speed up the read processing. 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

2.8 Pipeline

2.8.1 Overview

The RX CPU has 5-stage pipeline structure. The RX CPU instruction is converted into one or more micro-operations, which are then executed in pipeline processing. In the pipeline stage, the IF stage is executed in the unit of instructions, while the D and subsequent stages are executed in the unit of micro-operations.
The operation of pipeline and respective stages is described below.

(1) IF stage (instruction fetch stage)

In the IF stage, the CPU fetches instructions from the memory. As the RX CPU has four 8-byte instruction queues, it fetches instructions until the instruction queue is full, regardless of the completion of decoding in the D (decoding) stage.

(2) D stage (decoding stage)

The CPU decodes instructions in the D stage and converts them into micro-operations. The CPU reads the register information (RF) in this stage and executes a bypass process (BYP) if the result of the preceding instruction will be used in a subsequent instruction. The write of operation result to the register (RW) can be executed with the register reference by using the bypass process.

(3) E stage (execution stage)

Operations and address calculations (OP) are processed in the E stage.

(4) M stage (memory access stage)

Operand memory accesses (OA1, OA2) are processed in the M stage. This stage is used only when the memory is accessed, and is divided into two sub-stages, M1 and M2. The RX CPU enables respective memory accesses for M1 and M2.

- M1 stage (memory-access stage 1)
  Operand memory access (OA1) is processed.
  Store operation: The pipeline processing ends when a write request is received via the bus.
  Load operation: The operation proceeds to the M2 stage when a read request is received via the bus. If a request and load data are received at the same timing (no-wait memory access), the operation proceeds to the WB stage.
- M2 stage (memory-access stage 2)
  Operand memory access (OA2) is processed. The CPU waits for the load data in the M2 stage. When the load data is received, the operation proceeds to the WB stage.

(5) WB stage (write-back stage)

The operation result and the data read from memory are written to the register (RW) in the WB stage. The data read from memory and the other type of data, such as the operation result, can be written to the register in the same clock cycles.

Figure 2.10 shows the pipeline configuration and its operation.

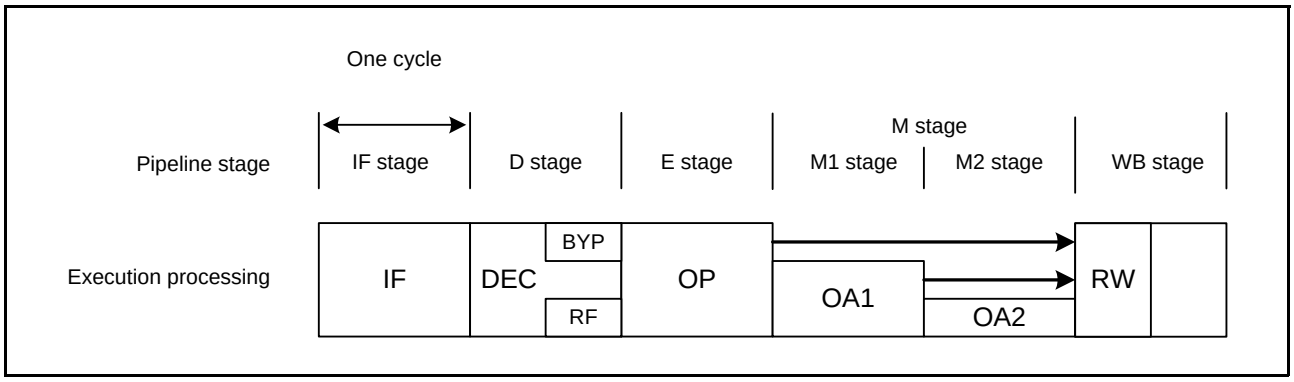


Figure 2.10 Pipeline Configuration and its Operation

2.8.2 Instructions and Pipeline Processing

The operands in the table below indicate the following meaning.

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

2.8.2.1 Instructions Converted into Single Micro-Operation and Pipeline Processing

The table below lists the instructions that are converted into a single micro-operation. The number of cycles in the table indicates the number of cycles during no-wait memory access.

Table 2.13 Instructions that are Converted into a Single Micro-Operation

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
Arithmetic/logic instructions (register-register, immediate-register) Except EMUL, EMULU, RMPA, DIV, DIVU and SATR	• {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”	Figure 2.11	1
Arithmetic/logic instructions (division)	• DIV “#IMM, Rd”/“Rs, Rd”	Figure 2.11	3 to 20*1
	• DIVU “#IMM, Rd”/“Rs, Rd”	Figure 2.11	2 to 18*1
Data transfer instructions (register-register, immediate-register)	• MOV “#IMM, Rd”/“Rs, Rd” • {MOVU, REVL, REVW} “Rs, Rd” • SC*Cnd* “Rd” • {STNZ, STZ} “#IMM, Rd”	Figure 2.11	1
Transfer instructions (load operation)	• {MOV, MOVU} “[Rs], Rd”/“dsp[Rs], Rd”/“[Rs+], Rd”/“[-Rs], Rd”/“[Ri, Rb], Rd” • POP “Rd”	Figure 2.12	Throughput: 1 Latency: 2*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]”	Figure 2.13	1
Bit manipulation instructions (register)	• {BCLR, BNOT, BSET} “#IMM, Rd”/“Rs, Rd” • BM*Cnd* “#IMM, Rd” • BTST “#IMM, Rs”/“Rs, Rs2”	Figure 2.11	1
Branch instructions	• B*Cnd* “pcdsp” • {BRA, BSR} “pcdsp”/“Rs” • {JMP, JSR} “Rs”	Figure 2.22	Branch taken: 3 Branch not taken: 1
Floating-point operation instructions (register-register, immediate-register)	• FCMP “#IMM, Rd”/“Rs, Rs2”	Figure 2.11	1
System manipulation instructions	• {CLRPSW, SETPSW} “flag” • MVTC “#IMM, CR”/“Rs, CR” • MVFC “CR, Rd” • MVTIPL“#IMM”	—	1
DSP instructions	• {MACHI, MACLO, MULHI, MULLO} “Rs, Rs2” • {MVFACHI, MVFACMI} “Rd” • {MVTACHI, MVTACLO} “Rs” • RACW“#IMM”	Figure 2.11	1

Note 1. The number of cycles for the dividing instruction varies according to the divisor and dividend.
Note 2. For the number of cycles for throughput and latency, see section 2.8.3, Calculation of the Instruction Processing Time.

Figure 2.11 to Figure 2.13 show the operation of instructions that are converted into a basic single micro-operation.

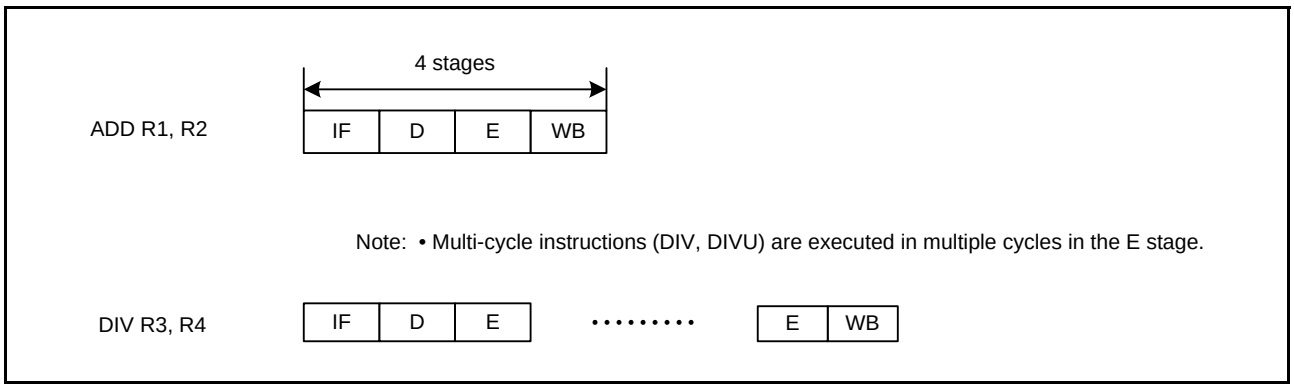


Figure 2.11 Operation for Register-Register, Immediate-Register

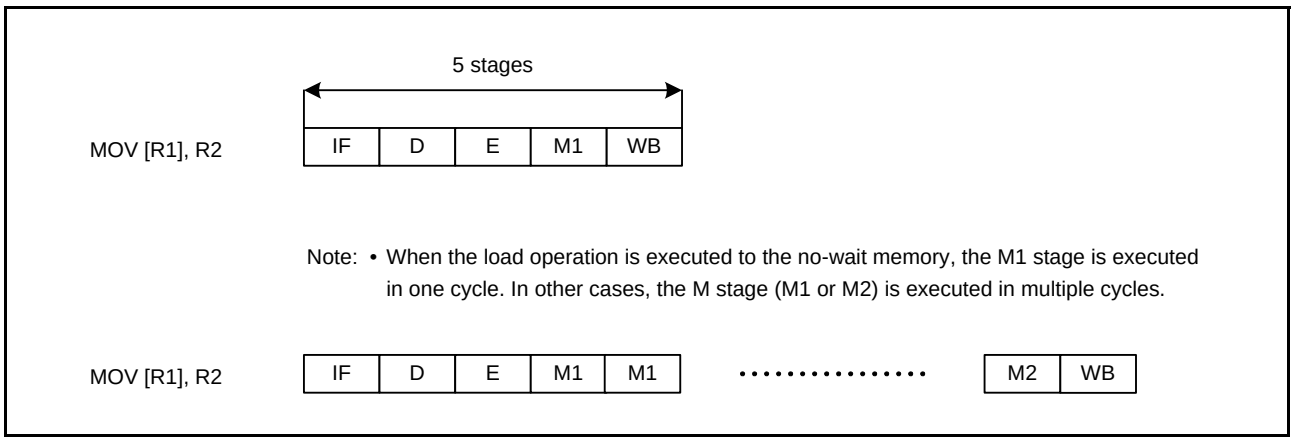


Figure 2.12 Load Operation

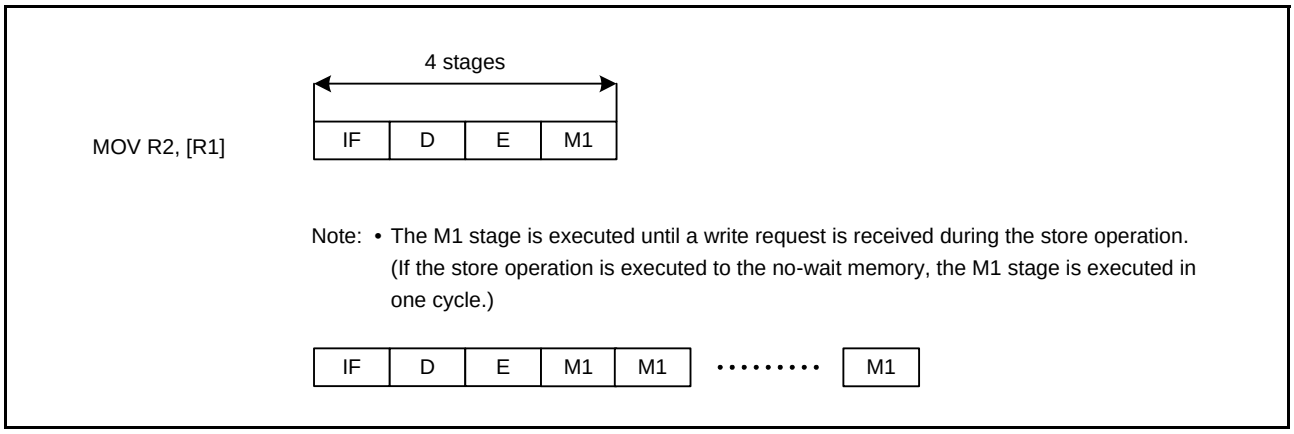


Figure 2.13 Store Operation

2.8.2.2 Instructions Converted into Multiple Micro-Operations and Pipeline Processing

The table below lists the instructions that are converted into multiple micro-operations. The number of cycles in the table indicates the number of cycles during no-wait memory access.

Table 2.14 Instructions that are Converted into Multiple Micro-Operations (1/2)

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
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”	Figure 2.14	3
Arithmetic/logic instructions (division)	• DIV “[Rs],Rd / dsp[Rs],Rd”	—	5 to 22
	• DIVU“[Rs],Rd / dsp[Rs],Rd”	—	4 to 20
Arithmetic/logic instructions (multiplier: 32 × 32 → 64 bits) (register-register, register-immediate)	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	Figure 2.16	2
Arithmetic/logic instructions (multiplier: 32 × 32 → 64 bits) (memory source operand)	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	—	4
Arithmetic/logic instructions (multiply-and-accumulate operation)	• RMPA.B	—	6+7×floor(n/4)+4×(n%4) n: Number of processing bytes*1
	• RMPA.W	—	6+5×floor(n/2)+4×(n%2) n: Number of processing words*1
	• RMPA.L	—	6+4n n: Number of processing longwords*1
Arithmetic/logic instructions (64-bit signed saturation processing for the RMPA instruction)	• SATR	—	3
Data transfer instructions (memory-memory transfer)	• MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]” • PUSH “[Rs]”/“dsp[Rs]”	Figure 2.15	3
Bit manipulation instructions (memory source operand)	• {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]” • BMCnd “#IMM, [Rd]”/“#IMM, dsp[Rd]” • BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2]	Figure 2.15	3
Transfer instructions (load operation)	• POPC “CR”	—	Throughput: 3 Latency: 4*2
Transfer instructions (save operation of multiple registers)	• PUSHM “Rs-Rs2”	—	n n: Number of registers*3
Transfer instructions (restore operation of multiple registers)	• POPM “Rs-Rs2”	—	Throughput: n Latency: n + 1 n: Number of registers*2,*4
Transfer instructions (register-register)	• XCHG “Rs, Rd”	Figure 2.17	2
Transfer instructions (memory-register)	• XCHG “[Rs], Rd”/“dsp[Rs], Rd”	Figure 2.18	2
Branch instructions	• 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*2

Table 2.14 Instructions that are Converted into Multiple Micro-Operations (2/2)

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
String manipulation instructions*5	• SCMPU	—	2+4×floor(n/4)+4×(n%4) n: Number of comparison bytes*1
	• SMOVB	—	n>3? 6+3×floor(n/4)+3×(n%4): 2+3n n: Number of transfer bytes*1
	• SMOVF, SMOVU	—	2+3×floor(n/4)+3×(n%4) n: Number of transfer bytes*1
	• SSTR.B	—	2+floor(n/4)+n%4 n: Number of transfer bytes*1
	• SSTR.W	—	2+floor(n/2)+n%2 n: Number of transfer words*1
	• 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*1
	• SUNTIL.W, SWHILE.W	—	3+3×floor(n/2)+3×(n%2) n: Number of comparison words*1
	• SUNTIL.L, SWHILE.L	—	3+3×n n: Number of comparison longwords
Floating-point operation instructions (register-register, immediate-register)	• {FADD, FSUB} “#IMM, Rd”/“Rs, Rd”	Figure 2.19	4
	• FMUL “#IMM, Rd”/“Rs, Rd”	—	3
	• FDIV “#IMM, Rd”/“Rs, Rd”	—	16
	• {FTOI, ROUND, ITOF} “Rs, Rd”	—	2
Floating-point operation instructions (memory source operand)	• {FADD, FSUB} “[Rs], Rd”/“dsp[Rs], Rd”	—	6
	• FCMP “[Rs], Rs2”/“dsp[Rs], Rs2”	—	3
	• FMUL “[Rs], Rd”/“dsp[Rs], Rd”	—	5
	• FDIV “[Rs], Rd”/“dsp[Rs], Rd”	—	18
	• {FTOI, ROUND, ITOF} “[Rs], Rd”/ “dsp[Rs], Rd”	—	4
System manipulation instructions	• RTE	—	6
	• RTFI	—	3

?: Conditional operator

Note 1. floor(x): Max. integer that is smaller than x
Note 2. For the number of cycles for throughput and latency, see section 2.8.3, Calculation of the Instruction Processing Time.
Note 3. The PUSHM instruction is converted into multiple store operations. The pipeline 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.
Note 4. The POPM instruction is converted into multiple load operations. The pipeline 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 5. 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.

Figure 2.14 to Figure 2.21 show the operation of instructions that are converted into basic multiple micro-operations.
Note: • mop: Micro-operation, stall: Pipeline stall

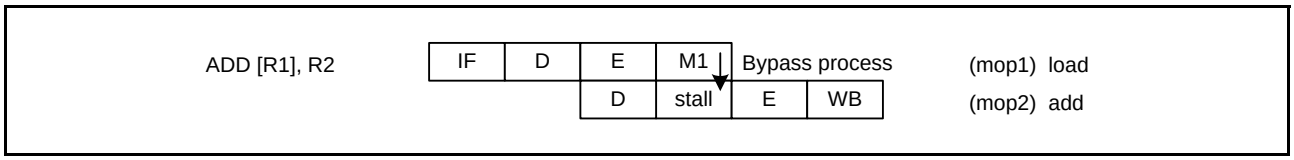


Figure 2.14 Arithmetic/Logic Instruction (Memory Source Operand)

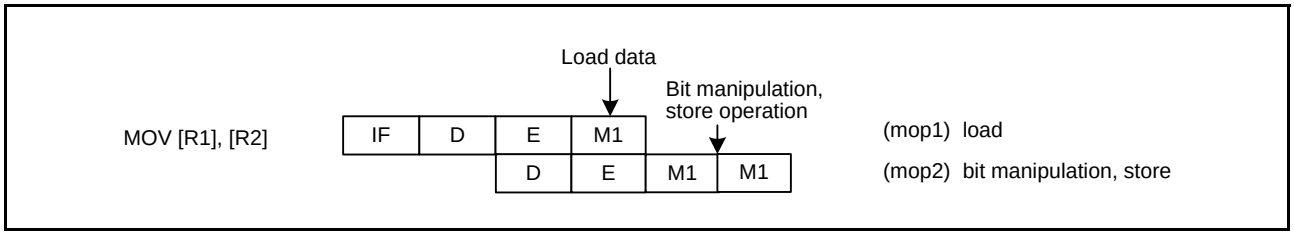


Figure 2.15 MOV Instruction (Memory-Memory), Bit Manipulation Instruction (Memory Source Operand)

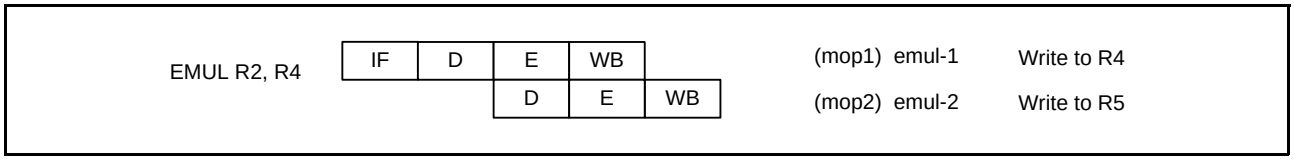


Figure 2.16 EMUL, EMULU Instructions (Register- Register, Register-Immediate)

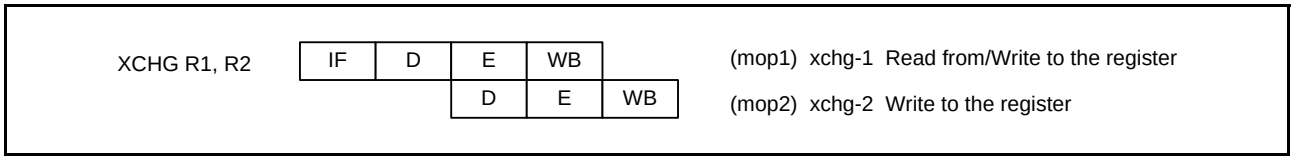


Figure 2.17 XCHG Instruction (Registers)

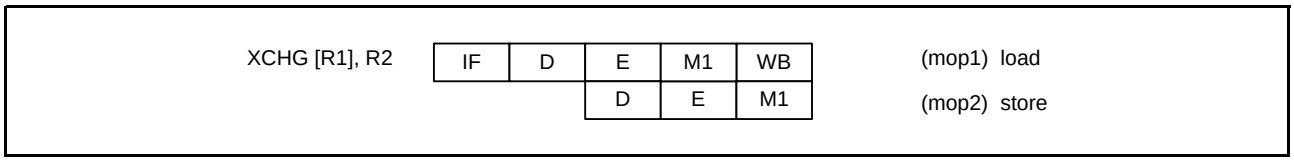


Figure 2.18 XCHG Instruction (Memory Source Operand)

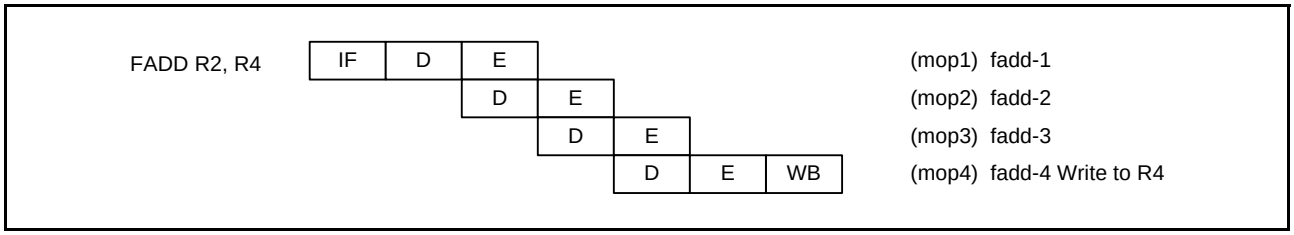


Figure 2.19 Floating-Point Operation Instruction (Register-Register, Immediate-Register)

2.8.2.3 Pipeline Basic Operation

In the ideal pipeline processing, each stage is executed in one cycle, though all instructions may not be pipelined in due to the processing in each stage and the branch execution.
The CPU controls the pipeline stage with the IF stage in the unit of instructions, while the D and subsequent stages in the unit of micro-operations.
The figures below show the pipeline processing of typical cases.
Note: • mop: Micro-operation, stall: Pipeline stall

(1) Pipeline Flow with Stalls

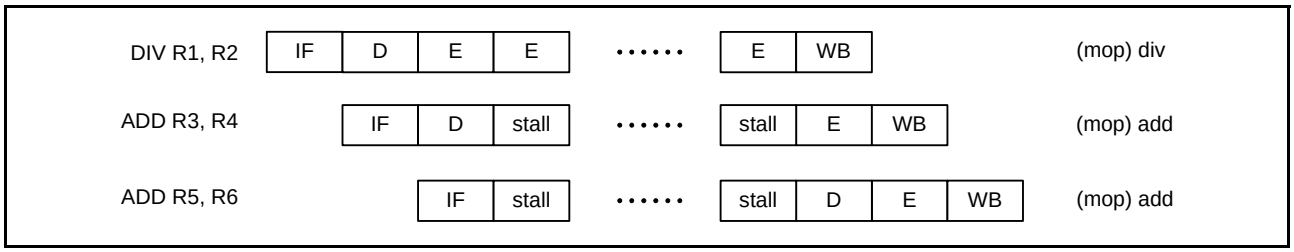


Figure 2.20 When an Instruction which Requires Multiple Cycles is Executed in the E Stage

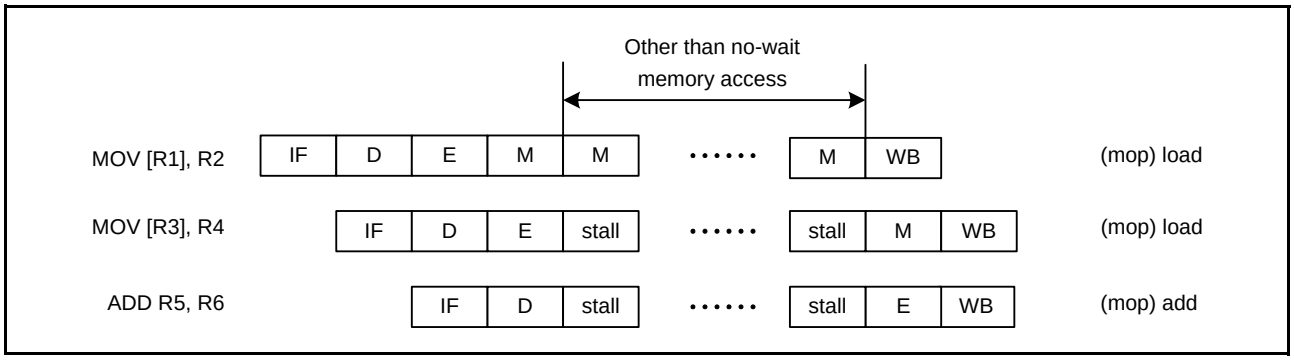


Figure 2.21 When an Instruction which Requires more than One Cycle for its Operand Access is Executed

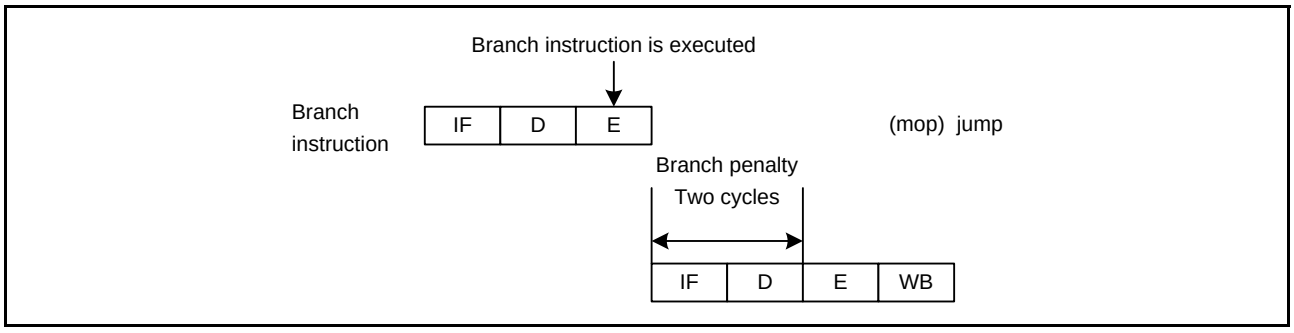


Figure 2.22 When a Branch Instruction is Executed (an Unconditional Branch Instruction is Executed or the Condition is Satisfied for a Conditional Branch Instruction)

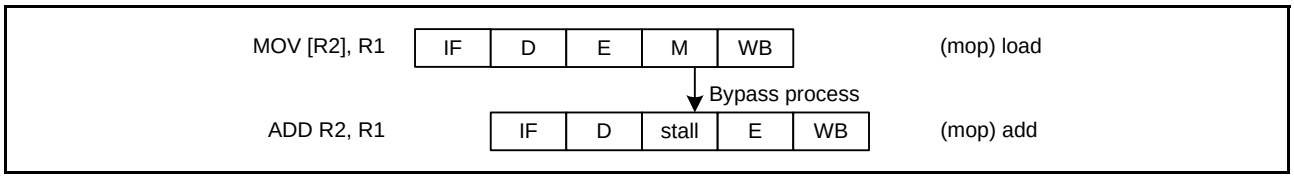


Figure 2.23 When the Subsequent Instruction Uses an Operand Read from the Memory

(2) Pipeline Flow with no Stall

(a) Bypass process

Even when the result of the preceding instruction will be used in a subsequent instruction, the operation processing between registers is pipelined in by the bypass process.

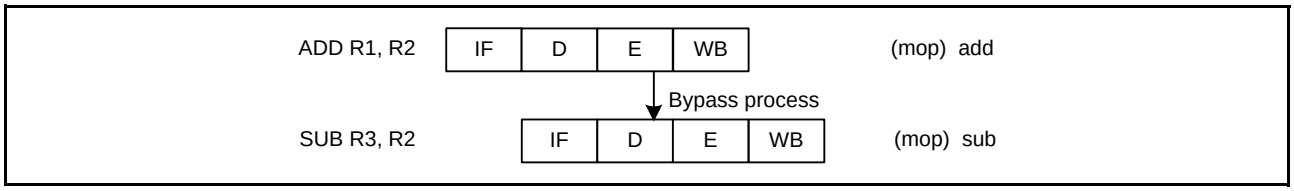


Figure 2.24 Bypass Process

(b) When WB stages for the memory load and for the operation are overlapped

Even when the WB stages for the memory load and for the operation are overlapped, the operation processing is pipelined in, because the load data and the operation result can be written to the register at the same timing.

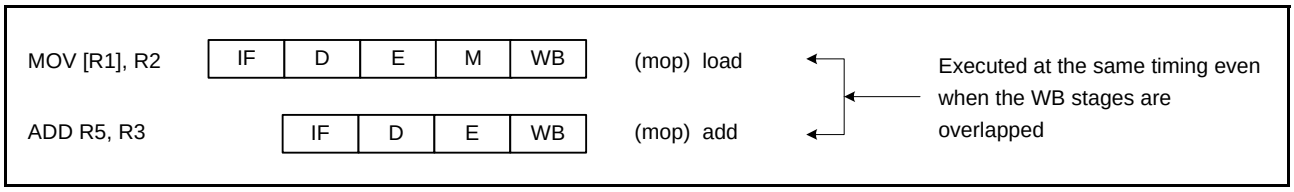


Figure 2.25 When WB Stages for the Memory Load and for the Operation are Overlapped

(c) When subsequent instruction writes to the same register before the end of memory load

Even when the subsequent instruction writes to the same register before the end of memory load, the operation processing is pipelined in, because the WB stage for the memory load is canceled.

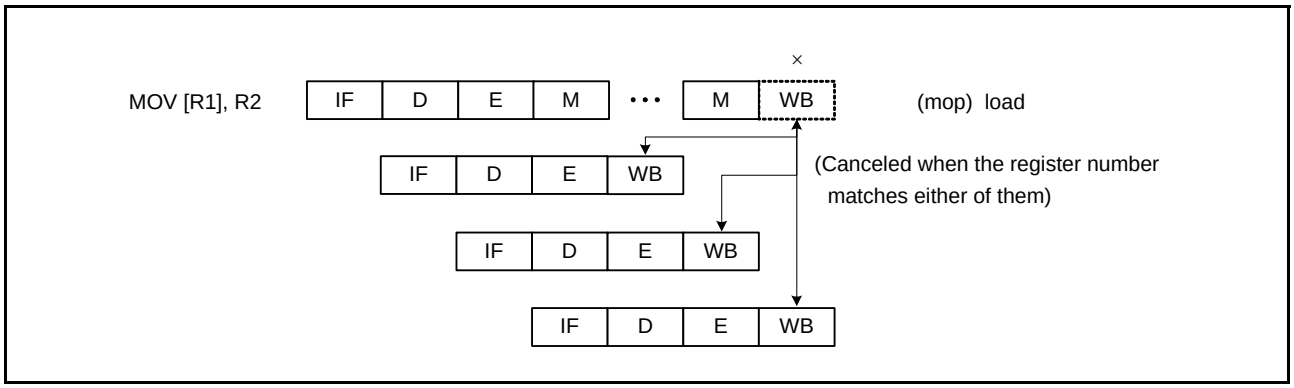


Figure 2.26 When Subsequent Instruction Writes to the Same Register before the End of Memory Load

(d) When the load data is not used by the subsequent instruction

When the load data is not used by the subsequent instruction, the subsequent operations are in fact executed earlier and the operation processing ends (out-of-order completion).

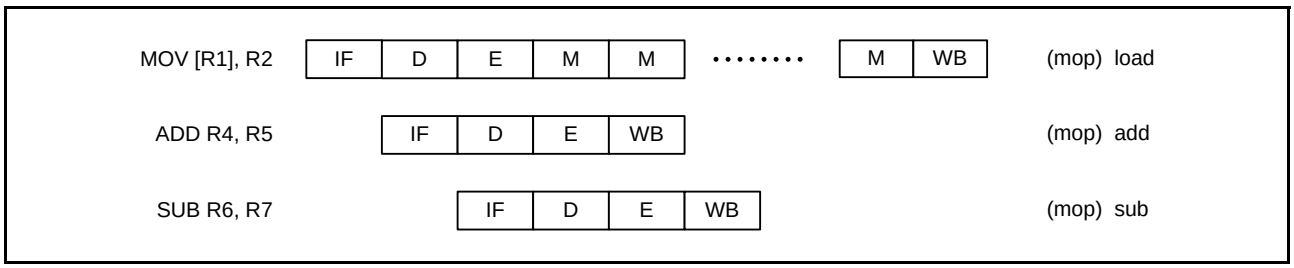


Figure 2.27 When Load Data is not Used by the Subsequent Instruction

2.8.3 Calculation of the Instruction Processing Time

Though the instruction processing time of the CPU varies according to the pipeline processing, the approximate time can be calculated in the following methods.

- Count the number of cycles (see Table 2.13 and Table 2.14)
- 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.
- If the instruction fetch stall is generated, the number of cycles increments.
- Depending on the system configuration, multiple cycles are required for the memory access.

2.8.4 Numbers of Cycles for Response to Interrupts

Table 2.15 lists numbers of cycles taken by processing for response to interrupts.

Table 2.15 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 2.15 will be applicable when access to memory from the CPU is processed with no waiting. The on-chip RAM and ROM in products of the RX63N/RX631 Groups allow such access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in on-chip ROM and the stack in on-chip RAM. Furthermore, place the addresses where the exception handling routine start on eight-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, Instructions that are Converted into a Single Micro-Operation, and Table 2.14, Instructions that are Converted into Multiple Micro-Operations.
The timing of interrupt acceptance depends on the state of the pipelines. 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

Operating modes are specified by the MD pin and the on-chip ROM enable (ROME) and external bus enable (EXBE) bits in the system control register 0 (SYSCR0).

Endian can be selected in each operating mode. Endian is specified by the endian select (MDE[2:0]) bits of the endian select registers (MDEB, MDES) in the option-setting memory. To use the user boot mode or USB boot mode, setting the UB code A and UB code B in the option-setting memory is required. For details on the option-setting memory, see section 7, Option-Setting Memory.

Do not change the state of the MD pin while the LSI is working. Do not select any combination other than those specified in Table 3.1.

Table 3.1 Selection of Operating Modes by the Mode Pin

Mode Pin	SYSCR0 Initial State				
MD	ROME	EXBE	Operating Mode	On-Chip ROM*1	External Bus
Low	1	0	Boot mode (SCI boot)	Enabled	Disabled
	1	0	User boot mode/USB boot mode*2	Enabled	Disabled
High	1	0	Single-chip mode	Enabled	Disabled

Note 1. The on-chip ROM is classified into two types: ROM and E2 DataFlash. For details, see section 45, ROM (Flash Memory for Code Storage), and section 46, E2 DataFlash Memory (Flash Memory for Data Storage).

Note 2. To be activated in user boot mode or USB boot mode, set the PC7 pin to high. Operation is in USB boot mode if overwriting of the user boot area is not required.

Table 3.2 Selection of Operating Modes by Register Setting

SYSCR0				
ROME	EXBE	Operating Mode	On-Chip ROM*1	External Bus
0	0	Single-chip mode, user boot mode	Disabled	Disabled
1	0		Enabled	Disabled
0	1	On-chip ROM disabled extended mode	Disabled	Enabled
1	1	On-chip ROM enabled extended mode	Enabled	Enabled

Note 1. The on-chip ROM is classified into two types: ROM and E2 DataFlash. For details, see section 45, ROM (Flash Memory for Code Storage), and section 46, E2 DataFlash Memory (Flash Memory for Data Storage).

Table 3.3 Endian Setting Method

Operating Mode	Endian Setting Method
User boot mode/USB boot mode	Endian is selected by the endian select registers (MDEB, MDES) in the option-setting memory.
Single-chip mode	

Note: • The endian select registers (MDEB, MDES) in the option-setting memory is allocated in the ROM.

Table 3.4 Selection of Endian

MDEB.MDE[2:0] Bits or MDES.MDE[2:0] Bits	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

Note 1. Depends on the setting of the mode pin (MD). When the MD pin is low, the bit value is 0; otherwise, the bit value is 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 and cannot be modified.	R
b8	—	Reserved	The read value is undefined. These bits cannot be modified.	R
b15 to b9	—	Reserved	These bits are read as 1 and cannot be modified.	R

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	BOTS	—	—	EXB	IROM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0/1*1	0/1*1	0	0	0	1

Note 1. Depends on the operating mode.

Bit	Symbol	Bit Name	Description	R/W
b0	IROM	On-Chip ROM Startup Status Flag	0: The on-chip ROM is disabled at startup. 1: The on-chip ROM is enabled at startup.	R
b1	EXB	External Bus Startup Status Flag	0: The external bus is disabled at startup. 1: The external bus is enabled at startup.	R
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b4	BOTS	Boot Mode (SCI Boot) Startup Flag	0: Started with a mode except boot mode (SCI boot). 1: Started with boot mode (SCI boot).	R
b5	UBTS	User Boot Mode (USB boot mode) Startup Flag	0: Started with a mode except user boot mode/USB boot mode. 1: Started with user boot mode/USB boot mode.	R
b15 to b6	—	Reserved	These bits are read as 0 and cannot be modified.	R

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 EXBE and ROME bits.	R/W

ROME Bit (On-Chip ROM Enable)

The ROME bit enables or disables the on-chip ROM (ROM, E2 data flash).
While this bit is 1, it can be cleared to 0. While this bit is 0, it cannot be set to 1. Once the on-chip ROM is disabled by clearing this bit to 0, the on-chip ROM can no longer be enabled with the ROME bit.
A 0 should not be written to this bit during access to the on-chip ROM. After writing a 0 to this bit to disable the on-chip ROM, make sure that the ROME bit has been changed to 0 before proceeding to the next processing.

EXBE Bit (External Bus Enable)

The EXBE bit enables or disables the external bus.
Write 0 to this bit while the external bus cycle is not performed. Furthermore, do not write 0 to this bit while a program is running in the external address space.
Be careful when disabling the external bus because the external bus and the internal bus are concurrently activated in some cases. When the EXBE bit is changed, the bus should be accessed after the change in the EXBE bit is completed.
When the EXBE bit is changed, the I/O port setting should also be changed simultaneously. For details, see section 21, Multi-Function Pin Controller (MPC).

KEY[7:0] Bits (SYSCR0 Key Code)

The KEY[7:0] bits enable or disable modifying SYSCR0.
When writing a value to the ROME or EXBE bit, write 5Ah to the KEY[7:0] bits simultaneously. If SYSCR0 is modified with a KEY[7:0] value other than 5Ah, the ROME and EXBE values remain unchanged.

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RAME
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	RAME	RAM Enable	0: The on-chip RAM is disabled. 1: The on-chip RAM is enabled.	R/W
b15 to b1	—	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 on-chip RAM.

The RAME bit is initialized to 1 after a reset is released.

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

Even when the RAME bit is cleared to 0, the on-chip RAM retains its value. To retain the value in the on-chip RAM, keep the specified RAM standby voltage (VRAM). For details, see section 49, Electrical Characteristics.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In this mode, the on-chip ROM is enabled or disabled, the external bus is disabled (SYSCR0.EXBE bit = 0), and all I/O ports can be used as input/output ports.
The on-chip ROM is enabled when this LSI is started. While the on-chip ROM is enabled (SYSCR0.ROME bit = 1), it can be disabled by clearing the SYSCR0.ROME bit to 0. While the on-chip ROM is disabled (SYSCR0.ROME bit = 0), it cannot be enabled by setting the SYSCR0.ROME bit to 1.
Setting the SYSCR0.EXBE bit to 1 causes a transition to on-chip ROM enabled extended mode or on-chip ROM disabled extended mode where the external bus is 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 is available as external extended mode (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 21, Multi-Function Pin Controller (MPC).
The external bus width can be changed by the setting of external bus width selection (CSnCR.BSIZE[1:0] bits (n = 0 to 7)). For details, see section 16, Buses.
Writing 0 to the SYSCR0.EXBE bit causes a transition to single-chip mode (on-chip ROM enabled).
Writing 0 to the SYSCR0.ROME bit 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 is available as external extended mode (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 21, Multi-Function Pin Controller (MPC).
The external bus width can be changed by the setting of external bus width selection (CSnCR.BSIZE[1:0] bits (n=0 to 7)). For details, see section 16, Buses.
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 causes a transition to single-chip mode (on-chip ROM disabled).

3.3.4 Boot Mode (SCI boot)

Boot mode (SCI boot) is provided for the ROM. This mode functions in the same manner as single-chip mode except for data program/erase to the ROM. For details, see section 45, ROM (Flash Memory for Code Storage), and section 46, E2 DataFlash Memory (Flash Memory for Data Storage).

3.3.5 User Boot Mode

User boot mode is provided for executing the user boot mode. This mode functions in the same manner as single-chip mode except for data program/erase to the ROM. For details, see section 45, ROM (Flash Memory for Code Storage), and section 46, E2 DataFlash Memory (Flash Memory for Data Storage).
Transition to software standby mode or deep software standby must not be made in user boot mode.

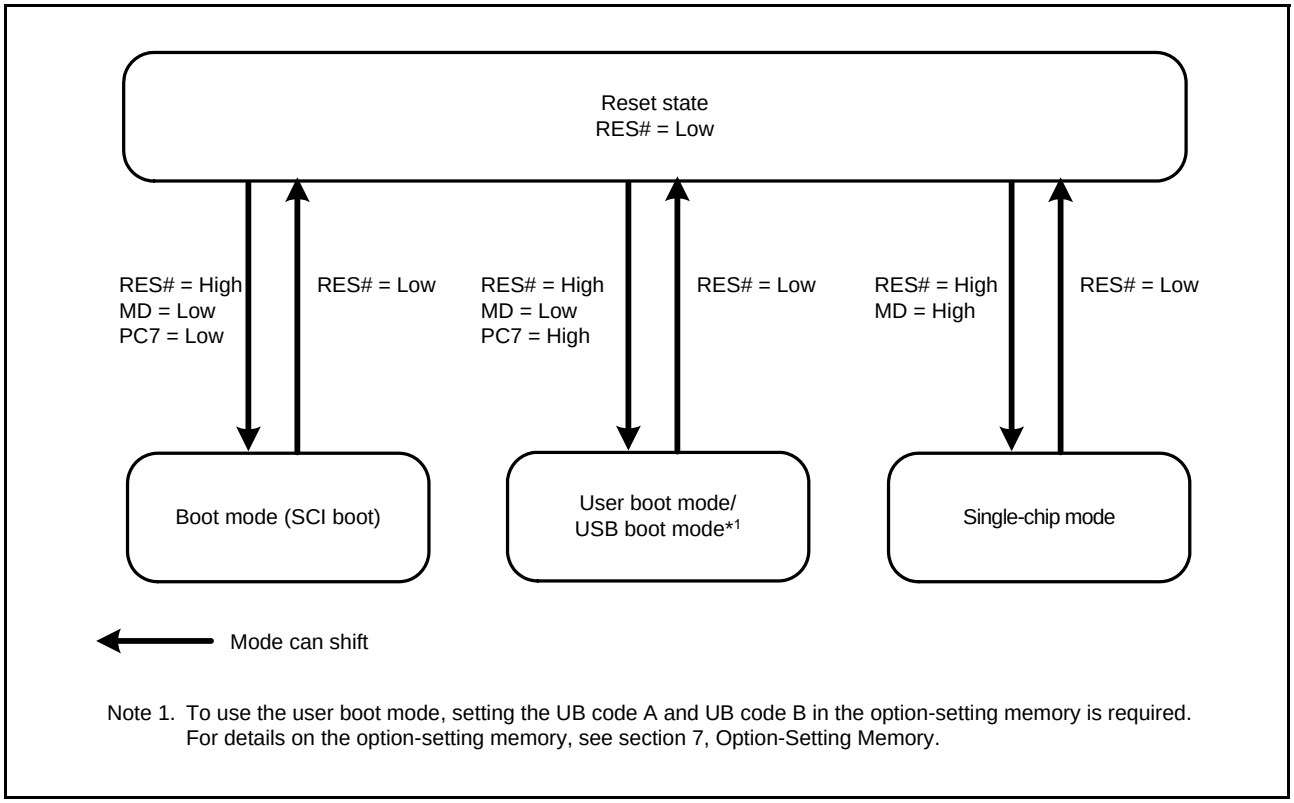
3.3.6 USB Boot Mode

USB boot mode is provided for the ROM. This mode functions in the same manner as single-chip mode except for data program/erase to the ROM. For details, see section 45, ROM (Flash Memory for Code Storage), and section 46, E2 DataFlash Memory (Flash Memory for Data Storage).

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions According to Mode Pin Setting

Figure 3.1 shows operating mode transitions according to the setting of the MD pin. Operating modes can shift in the direction of arrow.



Note 1. To use the user boot mode, setting the UB code A and UB code B in the option-setting memory is required. For details on the option-setting memory, see section 7, Option-Setting Memory.

Figure 3.1 Setting of MD Pin and Operating Modes

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. Operating modes can shift in the direction of arrow.

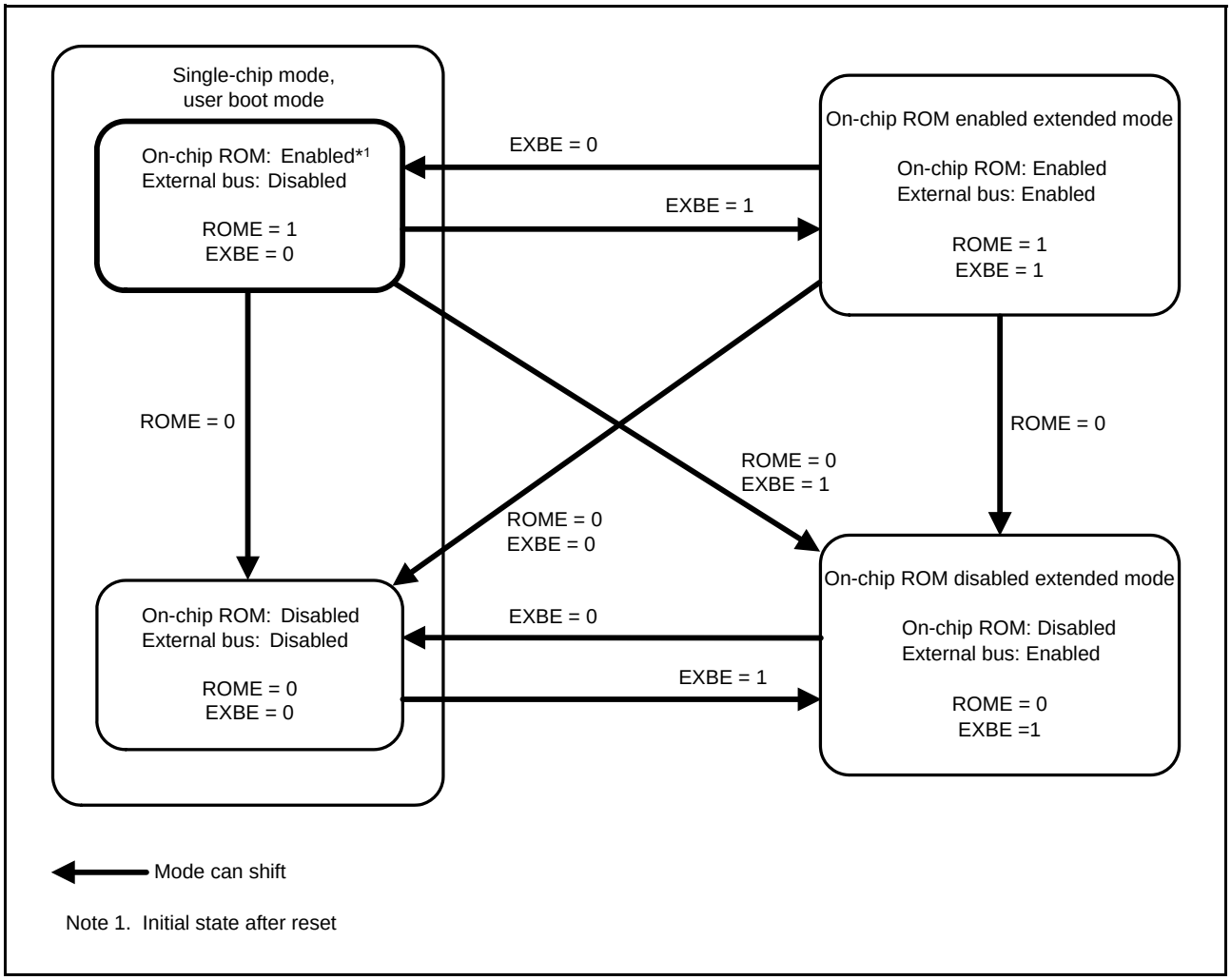
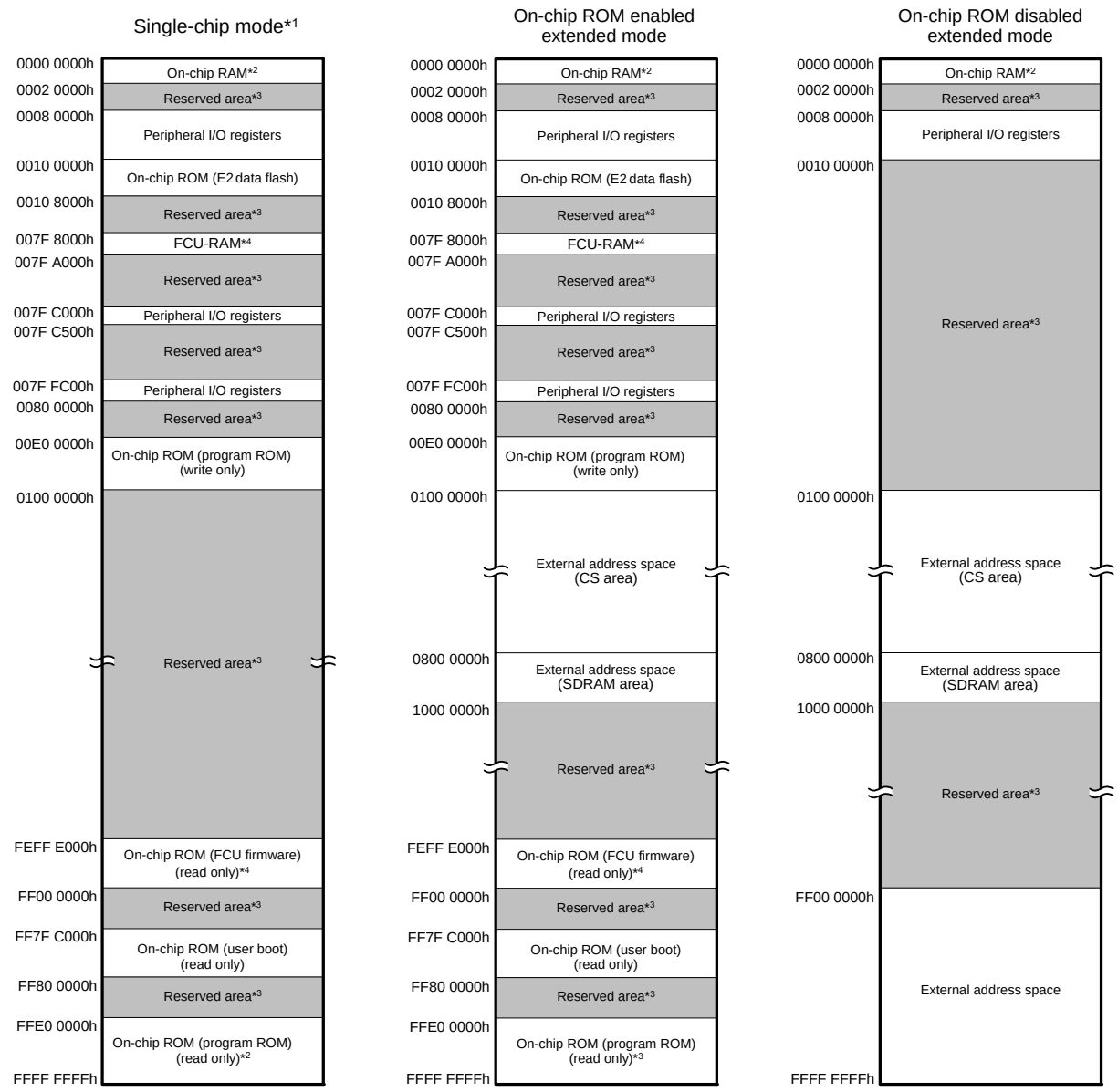


Figure 3.2 Setting of SYSCR0.ROME and EXBE Bits and Operating Modes

4. Address Space

4.1 Address Space

This LSI 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.

ROM (byt)			RAM (byt)	
Capacity	**Address (for reading only)**	**Address (for programming only)**	**Capacity**	**Address**
2 M	FFE0 0000h to FFFF FFFFh	00E0 0000h to 00FF FFFFh	128 K	0000 0000h～0001 FFFFh
1.5 M	FFE8 0000h to FFFF FFFFh	00E8 0000h to 00FF FFFFh		
1 M	FFF0 0000h to FFFF FFFFh	00F0 0000h to 00FF FFFFh		
768 K	FFF4 0000h to FFFF FFFFh	00F4 0000h to 00FF FFFFh		

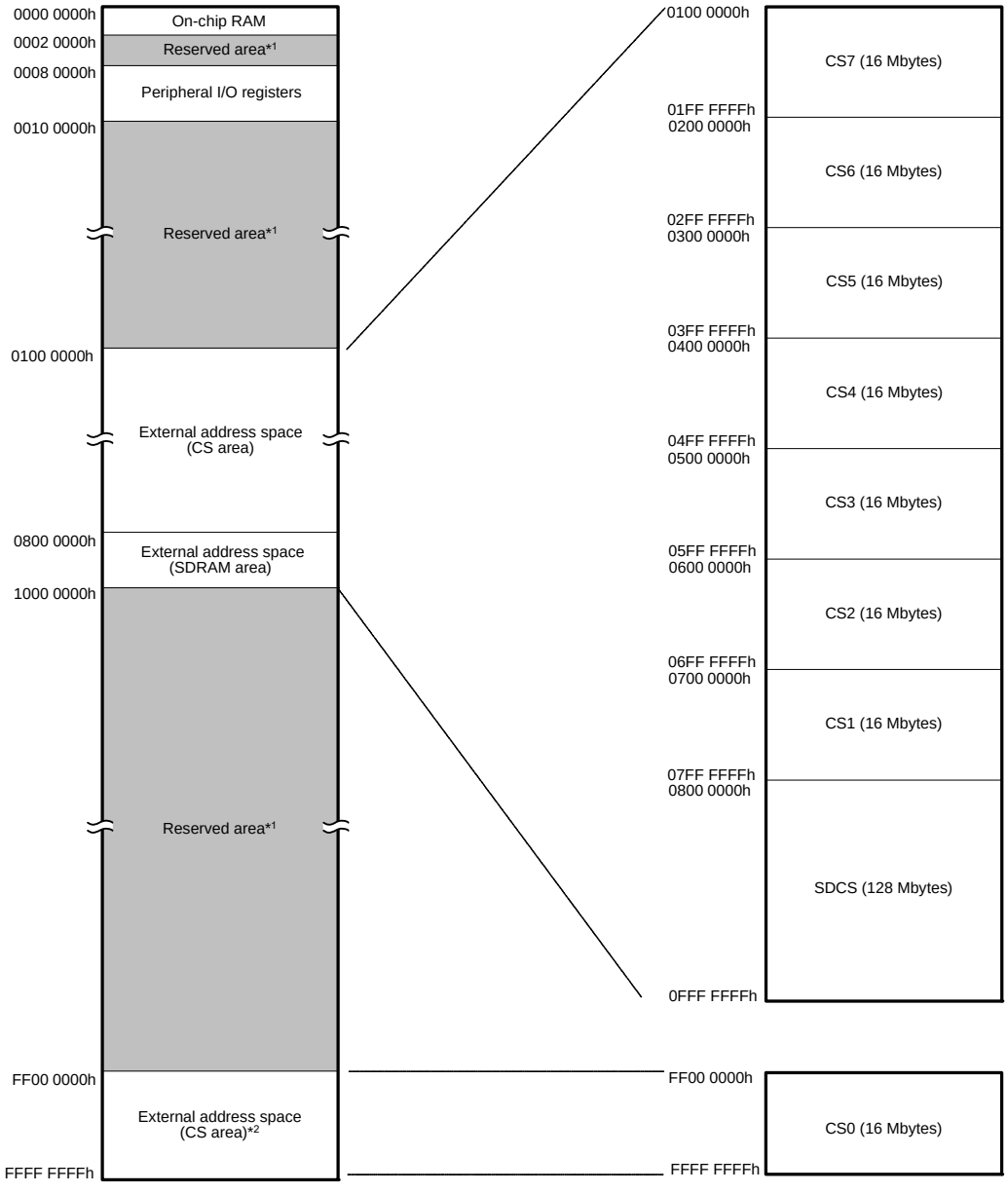
Note: • See Table 1.3, List of Products, for the product type name.

Note 3. Reserved areas should not be accessed.
Note 4. For details on the FCU, see section 45, ROM (Flash Memory for Code Storage) and section 46, E2 DataFlash Memory (Flash Memory for Data Storage).

Figure 4.1 Memory Map in Each Operating Mode

4.2 External Address Space

The external address space is classified into CS areas (CS0 to CS7) and SDRAM area (SDCS).
Figure 4.2 shows the address ranges corresponding to the individual CS areas (CS0 to CS7) and SDRAM area (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) cleared 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 bus 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).

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 0000h	SYSTEM	Mode monitor register	MDMONR	16	16	3 ICLK		section 3.
0008 0002h	SYSTEM	Mode status register	MDSR	16	16	3 ICLK		
0008 0006h	SYSTEM	System control register 0	SYSCR0	16	16	3 ICLK		
0008 0008h	SYSTEM	System control register 1	SYSCR1	16	16	3 ICLK		
0008 000Ch	SYSTEM	Standby control register	SBYCR	16	16	3 ICLK		section 11.
0008 0010h	SYSTEM	Module stop control register A	MSTPCRA	32	32	3 ICLK		
0008 0014h	SYSTEM	Module stop control register B	MSTPCRB	32	32	3 ICLK		
0008 0018h	SYSTEM	Module stop control register C	MSTPCRC	32	32	3 ICLK		
0008 0020h	SYSTEM	System clock control register	SCKCR	32	32	3 ICLK		section 9.
0008 0024h	SYSTEM	System clock control register 2	SCKCR2	16	16	3 ICLK		
0008 0026h	SYSTEM	System clock control register 3	SCKCR3	16	16	3 ICLK		
0008 0028h	SYSTEM	PLL control register	PLLCR	16	16	3 ICLK		
0008 002Ah	SYSTEM	PLL control register 2	PLLCR2	8	8	3 ICLK		
0008 0030h	SYSTEM	External bus clock control register	BCKCR	8	8	3 ICLK		
0008 0032h	SYSTEM	Main clock oscillator control register	MOSCCR	8	8	3 ICLK		
0008 0033h	SYSTEM	Sub-clock oscillator control register	SOSCCR	8	8	3 ICLK		
0008 0034h	SYSTEM	Low-speed on-chip oscillator control register	LOCOCR	8	8	3 ICLK		
0008 0035h	SYSTEM	IWDT-dedicated on-chip oscillator control register	ILOCOCR	8	8	3 ICLK		
0008 0036h	SYSTEM	High-speed on-chip oscillator control register	HOCOCR	8	8	3 ICLK		
0008 0040h	SYSTEM	Oscillation stop detect control register	OSTDCR	8	8	3 ICLK		
0008 0041h	SYSTEM	Oscillation stop detection status register	OSTDSR	8	8	3 ICLK		
0008 00A0h	SYSTEM	Operating power control register	OPCCR	8	8	3 ICLK		
0008 00A1h	SYSTEM	Sleep mode recovery clock source switching register	RSTCKCR	8	8	3 ICLK		section 11.
0008 00A2h	SYSTEM	Main clock oscillator wait control register	MOSCWTCR	8	8	3 ICLK		
0008 00A3h	SYSTEM	Sub-clock oscillator wait control register	SOSCWTCR	8	8	3 ICLK		
0008 00A6h	SYSTEM	PLL wait control register	PLLWTCR	8	8	3 ICLK		
0008 00C0h	SYSTEM	Reset status register 2	RSTSR2	8	8	3 ICLK		section 6.
0008 00C2h	SYSTEM	Software reset register	SWRR	16	16	3 ICLK		
0008 00E0h	SYSTEM	Voltage monitoring 1 circuit control register 1	LVD1CR1	8	8	3 ICLK		section 8.
0008 00E1h	SYSTEM	Voltage monitoring 1 circuit status register	LVD1SR	8	8	3 ICLK		
0008 00E2h	SYSTEM	Voltage monitoring 2 circuit control register 1	LVD2CR1	8	8	3 ICLK		
0008 00E3h	SYSTEM	Voltage monitoring 2 circuit status register	LVD2SR	8	8	3 ICLK		
0008 03FEh	SYSTEM	Protection register	PRCR	16	16	3 ICLK		section 13.
0008 1300h	BSC	Bus error status clear register	BERCLR	8	8	2 ICLK		section 16.
0008 1304h	BSC	Bus error monitoring enable register	BEREN	8	8	2 ICLK		
0008 1308h	BSC	Bus error status register 1	BERSR1	8	8	2 ICLK		
0008 130Ah	BSC	Bus error status register 2	BERSR2	16	16	2 ICLK		
0008 1310h	BSC	Bus priority control register	BUSPRI	16	16	2 ICLK		
0008 2000h	DMAC0	DMA source address register	DMSAR	32	32	2 ICLK		section 17.
0008 2004h	DMAC0	DMA source address register	DMDAR	32	32	2 ICLK		
0008 2008h	DMAC0	DMA transfer count register	DMCRA	32	32	2 ICLK		
0008 200Ch	DMAC0	DMA block transfer count register	DMCRB	16	16	2 ICLK		
0008 2010h	DMAC0	DMA transfer mode register	DMTMD	16	16	2 ICLK		
0008 2013h	DMAC0	DMA interrupt setting register	DMINT	8	8	2 ICLK		
0008 2014h	DMAC0	DMA address mode register	DMAMD	16	16	2 ICLK		
0008 2018h	DMAC0	DMA offset register	DMOFR	32	32	2 ICLK		
0008 201Ch	DMAC0	DMA transfer enable register	DMCNT	8	8	2 ICLK		
0008 201Dh	DMAC0	DMA software start register	DMREQ	8	8	2 ICLK		
0008 201Eh	DMAC0	DMA status register	DMSTS	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (2/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 201Fh	DMAC0	DMA activation source flag control register	DMCSL	8	8	2 ICLK		section 17.
0008 2040h	DMAC1	DMA source address register	DMSAR	32	32	2 ICLK		
0008 2044h	DMAC1	DMA source address register	DMDAR	32	32	2 ICLK		
0008 2048h	DMAC1	DMA transfer count register	DMCRA	32	32	2 ICLK		
0008 204Ch	DMAC1	DMA block transfer count register	DMCRB	16	16	2 ICLK		
0008 2050h	DMAC1	DMA transfer mode register	DMTMD	16	16	2 ICLK		
0008 2053h	DMAC1	DMA interrupt setting register	DMINT	8	8	2 ICLK		
0008 2054h	DMAC1	DMA address mode register	DMAMD	16	16	2 ICLK		
0008 205Ch	DMAC1	DMA transfer enable register	DMCNT	8	8	2 ICLK		
0008 205Dh	DMAC1	DMA software start register	DMREQ	8	8	2 ICLK		
0008 205Eh	DMAC1	DMA status register	DMSTS	8	8	2 ICLK		
0008 205Fh	DMAC1	DMA activation source flag control register	DMCSL	8	8	2 ICLK		
0008 2080h	DMAC2	DMA source address register	DMSAR	32	32	2 ICLK		
0008 2084h	DMAC2	DMA source address register	DMDAR	32	32	2 ICLK		
0008 2088h	DMAC2	DMA transfer count register	DMCRA	32	32	2 ICLK		
0008 208Ch	DMAC2	DMA block transfer count register	DMCRB	16	16	2 ICLK		
0008 2090h	DMAC2	DMA transfer mode register	DMTMD	16	16	2 ICLK		
0008 2093h	DMAC2	DMA interrupt setting register	DMINT	8	8	2 ICLK		
0008 2094h	DMAC2	DMA address mode register	DMAMD	16	16	2 ICLK		
0008 209Ch	DMAC2	DMA transfer enable register	DMCNT	8	8	2 ICLK		
0008 209Dh	DMAC2	DMA software start register	DMREQ	8	8	2 ICLK		
0008 209Eh	DMAC2	DMA status register	DMSTS	8	8	2 ICLK		
0008 209Fh	DMAC2	DMA activation source flag control register	DMCSL	8	8	2 ICLK		
0008 20C0h	DMAC3	DMA source address register	DMSAR	32	32	2 ICLK		
0008 20C4h	DMAC3	DMA source address register	DMDAR	32	32	2 ICLK		
0008 20C8h	DMAC3	DMA transfer count register	DMCRA	32	32	2 ICLK		
0008 20CCh	DMAC3	DMA block transfer count register	DMCRB	16	16	2 ICLK		
0008 20D0h	DMAC3	DMA transfer mode register	DMTMD	16	16	2 ICLK		
0008 20D3h	DMAC3	DMA interrupt setting register	DMINT	8	8	2 ICLK		
0008 20D4h	DMAC3	DMA address mode register	DMAMD	16	16	2 ICLK		
0008 20DCh	DMAC3	DMA transfer enable register	DMCNT	8	8	2 ICLK		
0008 20DDh	DMAC3	DMA software start register	DMREQ	8	8	2 ICLK		
0008 20DEh	DMAC3	DMA status register	DMSTS	8	8	2 ICLK		
0008 20DFh	DMAC3	DMA activation source flag control register	DMCSL	8	8	2 ICLK		
0008 2200h	DMAC	DMACA module activation register	DMAST	8	8	2 ICLK		
0008 2400h	DTC	DTC control register	DTCCR	8	8	2 ICLK		section 19.
0008 2404h	DTC	DTC vector base register	DTCVBR	32	32	2 ICLK		
0008 2408h	DTC	DTC address mode register	DTCADMOD	8	8	2 ICLK		
0008 240Ch	DTC	DTC module start register	DTCST	8	8	2 ICLK		
0008 240Eh	DTC	DTC status register	DTCSTS	16	16	2 ICLK		
0008 2800h	EXDMAC0	EXDMA source address register	EDMSAR	32	32	1, 2 BCLK		section 18.
0008 2804h	EXDMAC0	EXDMA destination address register	EDMDAR	32	32	1, 2 BCLK		
0008 2808h	EXDMAC0	EXDMA transfer count register	EDMCRA	32	32	1, 2 BCLK		
0008 280Ch	EXDMAC0	EXDMA block transfer count register	EDMCRB	16	16	1, 2 BCLK		
0008 2810h	EXDMAC0	EXDMA transfer mode register	EDMTMD	16	16	1, 2 BCLK		
0008 2812h	EXDMAC0	EXDMA output setting register	EDMOMD	8	8	1, 2 BCLK		
0008 2813h	EXDMAC0	EXDMA interrupt setting register	EDMINT	8	8	1, 2 BCLK		
0008 2814h	EXDMAC0	EXDMA address mode register	EDMAMD	32	32	1, 2 BCLK		
0008 2818h	EXDMAC0	EXDMA offset register	EDMOFR	32	32	1, 2 BCLK		
0008 281Ch	EXDMAC0	EXDMA transfer enable register	EDMCNT	8	8	1, 2 BCLK		
0008 281Dh	EXDMAC0	EXDMA software start register	EDMREQ	8	8	1, 2 BCLK		
0008 281Eh	EXDMAC0	EXDMA status register	EDMSTS	8	8	1, 2 BCLK		

Table 5.1 List of I/O Registers (Address Order) (3/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 2820h	EXDMAC0	EXDMA external request sense mode register	EDMRMD	8	8	1, 2 BCLK		section 18.
0008 2821h	EXDMAC0	EXDMA external request flag register	EDMERF	8	8	1, 2 BCLK		
0008 2822h	EXDMAC0	EXDMA peripheral request flag register	EDMPRF	8	8	1, 2 BCLK		
0008 2840h	EXDMAC1	EXDMA source address register	EDMSAR	32	32	1, 2 BCLK		
0008 2844h	EXDMAC1	EXDMA destination address register	EDMDAR	32	32	1, 2 BCLK		
0008 2848h	EXDMAC1	EXDMA transfer count register	EDMCRA	32	32	1, 2 BCLK		
0008 284Ch	EXDMAC1	EXDMA block transfer count register	EDMCRB	16	16	1, 2 BCLK		
0008 2850h	EXDMAC1	EXDMA transfer mode register	EDMTMD	16	16	1, 2 BCLK		
0008 2852h	EXDMAC1	EXDMA output setting register	EDMOMD	8	8	1, 2 BCLK		
0008 2853h	EXDMAC1	EXDMA interrupt setting register	EDMINT	8	8	1, 2 BCLK		
0008 2854h	EXDMAC1	EXDMA address mode register	EDMAMD	32	32	1, 2 BCLK		
0008 285Ch	EXDMAC1	EXDMA transfer enable register	EDMCNT	8	8	1, 2 BCLK		
0008 285Dh	EXDMAC1	EXDMA software start register	EDMREQ	8	8	1, 2 BCLK		
0008 285Eh	EXDMAC1	EXDMA status register	EDMSTS	8	8	1, 2 BCLK		
0008 2860h	EXDMAC1	EXDMA external request sense mode register	EDMRMD	8	8	1, 2 BCLK		
0008 2861h	EXDMAC1	EXDMA external request flag register	EDMERF	8	8	1, 2 BCLK		
0008 2862h	EXDMAC1	EXDMA peripheral request flag register	EDMPRF	8	8	1, 2 BCLK		
0008 2A00h	EXDMAC	EXDMA module start register	EDMAST	8	8	1, 2 BCLK		
0008 2BE0h	EXDMAC	Cluster buffer register 0	CLSBR0	32	32	1, 2 BCLK		
0008 2BE4h	EXDMAC	Cluster buffer register 1	CLSBR1	32	32	1, 2 BCLK		
0008 2BE8h	EXDMAC	Cluster buffer register 2	CLSBR2	32	32	1, 2 BCLK		
0008 2BECh	EXDMAC	Cluster buffer register 3	CLSBR3	32	32	1, 2 BCLK		
0008 2BF0h	EXDMAC	Cluster buffer register 4	CLSBR4	32	32	1, 2 BCLK		
0008 2BF4h	EXDMAC	Cluster buffer register 5	CLSBR5	32	32	1, 2 BCLK		
0008 2BF8h	EXDMAC	Cluster buffer register 6	CLSBR6	32	32	1, 2 BCLK		
0008 2BFCh	EXDMAC	Cluster buffer register 7	CLSBR7	32	32	1, 2 BCLK		
0008 3002h	BSC	CS0 mode register	CS0MOD	16	16	1, 2 BCLK		section 16.
0008 3004h	BSC	CS0 wait control register 1	CS0WCR1	32	32	1, 2 BCLK		
0008 3008h	BSC	CS0 wait control register 2	CS0WCR2	32	32	1, 2 BCLK		
0008 3012h	BSC	CS1 mode register	CS1MOD	16	16	1, 2 BCLK		
0008 3014h	BSC	CS1 wait control register 1	CS1WCR1	32	32	1, 2 BCLK		
0008 3018h	BSC	CS1 wait control register 2	CS1WCR2	32	32	1, 2 BCLK		
0008 3022h	BSC	CS2 mode register	CS2MOD	16	16	1, 2 BCLK		
0008 3024h	BSC	CS2 wait control register 1	CS2WCR1	32	32	1, 2 BCLK		
0008 3028h	BSC	CS2 wait control register 2	CS2WCR2	32	32	1, 2 BCLK		
0008 3032h	BSC	CS3 mode register	CS3MOD	16	16	1, 2 BCLK		
0008 3034h	BSC	CS3 wait control register 1	CS3WCR1	32	32	1, 2 BCLK		
0008 3038h	BSC	CS3 wait control register 2	CS3WCR2	32	32	1, 2 BCLK		
0008 3042h	BSC	CS4 mode register	CS4MOD	16	16	1, 2 BCLK		
0008 3044h	BSC	CS4 wait control register 1	CS4WCR1	32	32	1, 2 BCLK		
0008 3048h	BSC	CS4 wait control register 2	CS4WCR2	32	32	1, 2 BCLK		
0008 3052h	BSC	CS5 mode register	CS5MOD	16	16	1, 2 BCLK		
0008 3054h	BSC	CS5 wait control register 1	CS5WCR1	32	32	1, 2 BCLK		
0008 3058h	BSC	CS5 wait control register 2	CS5WCR2	32	32	1, 2 BCLK		
0008 3062h	BSC	CS6 mode register	CS6MOD	16	16	1, 2 BCLK		
0008 3064h	BSC	CS6 wait control register 1	CS6WCR1	32	32	1, 2 BCLK		
0008 3068h	BSC	CS6 wait control register 2	CS6WCR2	32	32	1, 2 BCLK		
0008 3072h	BSC	CS7 mode register	CS7MOD	16	16	1, 2 BCLK		
0008 3074h	BSC	CS7 wait control register 1	CS7WCR1	32	32	1, 2 BCLK		
0008 3078h	BSC	CS7 wait control register 2	CS7WCR2	32	32	1, 2 BCLK		
0008 3802h	BSC	CS0 control register	CS0CR	16	16	1, 2 BCLK		
0008 380Ah	BSC	CS0 recovery cycle register	CS0REC	16	16	1, 2 BCLK		

Table 5.1 List of I/O Registers (Address Order) (4/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 3812h	BSC	CS1 control register	CS1CR	16	16	1, 2 BCLK		section 16.
0008 381Ah	BSC	CS1 recovery cycle register	CS1REC	16	16	1, 2 BCLK		
0008 3822h	BSC	CS2 control register	CS2CR	16	16	1, 2 BCLK		
0008 382Ah	BSC	CS2 recovery cycle register	CS2REC	16	16	1, 2 BCLK		
0008 3832h	BSC	CS3 control register	CS3CR	16	16	1, 2 BCLK		
0008 383Ah	BSC	CS3 recovery cycle register	CS3REC	16	16	1, 2 BCLK		
0008 3842h	BSC	CS4 control register	CS4CR	16	16	1, 2 BCLK		
0008 384Ah	BSC	CS4 recovery cycle register	CS4REC	16	16	1, 2 BCLK		
0008 3852h	BSC	CS5 control register	CS5CR	16	16	1, 2 BCLK		
0008 385Ah	BSC	CS5 recovery cycle register	CS5REC	16	16	1, 2 BCLK		
0008 3862h	BSC	CS6 control register	CS6CR	16	16	1, 2 BCLK		
0008 386Ah	BSC	CS6 recovery cycle register	CS6REC	16	16	1, 2 BCLK		
0008 3872h	BSC	CS7 control register	CS7CR	16	16	1, 2 BCLK		
0008 387Ah	BSC	CS7 recovery cycle register	CS7REC	16	16	1, 2 BCLK		
0008 3880h	BSC	CS recovery cycle insertion enable register	CSRECEN	16	16	1, 2 BCLK		
0008 3C00h	BSC	SDC control register	SDCCR	8	8	1, 2 BCLK		
0008 3C01h	BSC	SDC mode register	SDCMOD	8	8	1, 2 BCLK		
0008 3C02h	BSC	SDRAM access mode register	SDAMOD	8	8	1, 2 BCLK		
0008 3C10h	BSC	SDRAM self-refresh control register	SDSELF	8	8	1, 2 BCLK		
0008 3C14h	BSC	SDRAM refresh control register	SDRFCR	16	16	1, 2 BCLK		
0008 3C16h	BSC	SDRAM auto-refresh control register	SDRFEN	8	8	1, 2 BCLK		
0008 3C20h	BSC	SDRAM initialization sequence control register	SDICR	8	8	1, 2 BCLK		
0008 3C24h	BSC	SDRAM initialization register	SDIR	16	16	1, 2 BCLK		
0008 3C40h	BSC	SDRAM address register	SDADR	8	8	1, 2 BCLK		
0008 3C44h	BSC	SDRAM timing register	SDTR	32	32	1, 2 BCLK		
0008 3C48h	BSC	SDRAM mode register	SDMOD	16	16	1, 2 BCLK		
0008 3C50h	BSC	SDRAM status register	SDSR	8	8	1, 2 BCLK		
0008 7010h	ICU	Interrupt request register 016	IR016	8	8	2 ICLK		section 15.
0008 7015h	ICU	Interrupt request register 021	IR021	8	8	2 ICLK		
0008 7017h	ICU	Interrupt request register 023	IR023	8	8	2 ICLK		
0008 701Bh	ICU	Interrupt request register 027	IR027	8	8	2 ICLK		
0008 701Ch	ICU	Interrupt request register 028	IR028	8	8	2 ICLK		
0008 701Dh	ICU	Interrupt request register 029	IR029	8	8	2 ICLK		
0008 701Eh	ICU	Interrupt request register 030	IR030	8	8	2 ICLK		
0008 701Fh	ICU	Interrupt request register 031	IR031	8	8	2 ICLK		
0008 7020h	ICU	Interrupt request register 032	IR032	8	8	2 ICLK		
0008 7021h	ICU	Interrupt request register 033	IR033	8	8	2 ICLK		
0008 7022h	ICU	Interrupt request register 034	IR034	8	8	2 ICLK		
0008 7023h	ICU	Interrupt request register 035	IR035	8	8	2 ICLK		
0008 7024h	ICU	Interrupt request register 036	IR036	8	8	2 ICLK		
0008 7025h	ICU	Interrupt request register 037	IR037	8	8	2 ICLK		
0008 7026h	ICU	Interrupt request register 038	IR038	8	8	2 ICLK		
0008 7027h	ICU	Interrupt request register 039	IR039	8	8	2 ICLK		
0008 7028h	ICU	Interrupt request register 040	IR040	8	8	2 ICLK		
0008 7029h	ICU	Interrupt request register 041	IR041	8	8	2 ICLK		
0008 702Ah	ICU	Interrupt request register 042	IR042	8	8	2 ICLK		
0008 702Bh	ICU	Interrupt request register 043	IR043	8	8	2 ICLK		
0008 702Ch	ICU	Interrupt request register 044	IR044	8	8	2 ICLK		
0008 702Dh	ICU	Interrupt request register 045	IR045	8	8	2 ICLK		
0008 702Eh	ICU	Interrupt request register 046	IR046	8	8	2 ICLK		
0008 702Fh	ICU	Interrupt request register 047	IR047	8	8	2 ICLK		
0008 7030h	ICU	Interrupt request register 048	IR048	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (5/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 7031h	ICU	Interrupt request register 049	IR049	8	8	2 ICLK		section 15.
0008 7032hz	ICU	Interrupt request register 050	IR050	8	8	2 ICLK		
0008 7033h	ICU	Interrupt request register 051	IR051	8	8	2 ICLK		
0008 7034h	ICU	Interrupt request register 052	IR052	8	8	2 ICLK		
0008 7035h	ICU	Interrupt request register 053	IR053	8	8	2 ICLK		
0008 7036h	ICU	Interrupt request register 054	IR054	8	8	2 ICLK		
0008 7037h	ICU	Interrupt request register 055	IR055	8	8	2 ICLK		
0008 7038h	ICU	Interrupt request register 056	IR056	8	8	2 ICLK		
0008 7039h	ICU	Interrupt request register 057	IR057	8	8	2 ICLK		
0008 703Ah	ICU	Interrupt request register 058	IR058	8	8	2 ICLK		
0008 703Bh	ICU	Interrupt request register 059	IR059	8	8	2 ICLK		
0008 703Eh	ICU	Interrupt request register 062	IR062	8	8	2 ICLK		
0008 7040h	ICU	Interrupt request register 064	IR064	8	8	2 ICLK		
0008 7041h	ICU	Interrupt request register 065	IR065	8	8	2 ICLK		
0008 7042h	ICU	Interrupt request register 066	IR066	8	8	2 ICLK		
0008 7043h	ICU	Interrupt request register 067	IR067	8	8	2 ICLK		
0008 7044h	ICU	Interrupt request register 068	IR068	8	8	2 ICLK		
0008 7045h	ICU	Interrupt request register 069	IR069	8	8	2 ICLK		
0008 7046h	ICU	Interrupt request register 070	IR070	8	8	2 ICLK		
0008 7047h	ICU	Interrupt request register 071	IR071	8	8	2 ICLK		
0008 7048h	ICU	Interrupt request register 072	IR072	8	8	2 ICLK		
0008 7049h	ICU	Interrupt request register 073	IR073	8	8	2 ICLK		
0008 704Ah	ICU	Interrupt request register 074	IR074	8	8	2 ICLK		
0008 704Bh	ICU	Interrupt request register 075	IR075	8	8	2 ICLK		
0008 704Ch	ICU	Interrupt request register 076	IR076	8	8	2 ICLK		
0008 704Dh	ICU	Interrupt request register 077	IR077	8	8	2 ICLK		
0008 704Eh	ICU	Interrupt request register 078	IR078	8	8	2 ICLK		
0008 704Fh	ICU	Interrupt request register 079	IR079	8	8	2 ICLK		
0008 705Ah	ICU	Interrupt request register 090	IR090	8	8	2 ICLK		
0008 705Bh	ICU	Interrupt request register 091	IR091	8	8	2 ICLK		
0008 705Ch	ICU	Interrupt request register 092	IR092	8	8	2 ICLK		
0008 705Dh	ICU	Interrupt request register 093	IR093	8	8	2 ICLK		
0008 7062h	ICU	Interrupt request register 098	IR098	8	8	2 ICLK		
0008 7066h	ICU	Interrupt request register 102	IR102	8	8	2 ICLK		
0008 706Ah	ICU	Interrupt request register 106	IR106	8	8	2 ICLK		
0008 706Bh	ICU	Interrupt request register 107	IR107	8	8	2 ICLK		
0008 706Ch	ICU	Interrupt request register 108	IR108	8	8	2 ICLK		
0008 706Dh	ICU	Interrupt request register 109	IR109	8	8	2 ICLK		
0008 706Eh	ICU	Interrupt request register 110	IR110	8	8	2 ICLK		
0008 706Fh	ICU	Interrupt request register 111	IR111	8	8	2 ICLK		
0008 7070h	ICU	Interrupt request register 112	IR112	8	8	2 ICLK		
0008 7072h	ICU	Interrupt request register 114	IR114	8	8	2 ICLK		
0008 707Ah	ICU	Interrupt request register 122	IR122	8	8	2 ICLK		
0008 707Bh	ICU	Interrupt request register 123	IR123	8	8	2 ICLK		
0008 707Ch	ICU	Interrupt request register 124	IR124	8	8	2 ICLK		
0008 707Dh	ICU	Interrupt request register 125	IR125	8	8	2 ICLK		
0008 707Eh	ICU	Interrupt request register 126	IR126	8	8	2 ICLK		
0008 707Fh	ICU	Interrupt request register 127	IR127	8	8	2 ICLK		
0008 7080h	ICU	Interrupt request register 128	IR128	8	8	2 ICLK		
0008 7081h	ICU	Interrupt request register 129	IR129	8	8	2 ICLK		
0008 7082h	ICU	Interrupt request register 130	IR130	8	8	2 ICLK		
0008 7083h	ICU	Interrupt request register 131	IR131	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (6/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 7084h	ICU	Interrupt request register 132	IR132	8	8	2 ICLK		section 15.
0008 7085h	ICU	Interrupt request register 133	IR133	8	8	2 ICLK		
0008 7086h	ICU	Interrupt request register 134	IR134	8	8	2 ICLK		
0008 7087h	ICU	Interrupt request register 135	IR135	8	8	2 ICLK		
0008 7088h	ICU	Interrupt request register 136	IR136	8	8	2 ICLK		
0008 7089h	ICU	Interrupt request register 137	IR137	8	8	2 ICLK		
0008 708Ah	ICU	Interrupt request register 138	IR138	8	8	2 ICLK		
0008 708Bh	ICU	Interrupt request register 139	IR139	8	8	2 ICLK		
0008 708Ch	ICU	Interrupt request register 140	IR140	8	8	2 ICLK		
0008 708Dh	ICU	Interrupt request register 141	IR141	8	8	2 ICLK		
0008 708Eh	ICU	Interrupt request register 142	IR142	8	8	2 ICLK		
0008 708Fh	ICU	Interrupt request register 143	IR143	8	8	2 ICLK		
0008 7090h	ICU	Interrupt request register 144	IR144	8	8	2 ICLK		
0008 7091h	ICU	Interrupt request register 145	IR145	8	8	2 ICLK		
0008 7092h	ICU	Interrupt request register 146	IR146	8	8	2 ICLK		
0008 7093h	ICU	Interrupt request register 147	IR147	8	8	2 ICLK		
0008 7094h	ICU	Interrupt request register 148	IR148	8	8	2 ICLK		
0008 7095h	ICU	Interrupt request register 149	IR149	8	8	2 ICLK		
0008 7096h	ICU	Interrupt request register 150	IR150	8	8	2 ICLK		
0008 7097h	ICU	Interrupt request register 151	IR151	8	8	2 ICLK		
0008 7098h	ICU	Interrupt request register 152	IR152	8	8	2 ICLK		
0008 7099h	ICU	Interrupt request register 153	IR153	8	8	2 ICLK		
0008 709Ah	ICU	Interrupt request register 154	IR154	8	8	2 ICLK		
0008 709Bh	ICU	Interrupt request register 155	IR155	8	8	2 ICLK		
0008 709Ch	ICU	Interrupt request register 156	IR156	8	8	2 ICLK		
0008 709Dh	ICU	Interrupt request register 157	IR157	8	8	2 ICLK		
0008 709Eh	ICU	Interrupt request register 158	IR158	8	8	2 ICLK		
0008 709Fh	ICU	Interrupt request register 159	IR159	8	8	2 ICLK		
0008 70A0h	ICU	Interrupt request register 160	IR160	8	8	2 ICLK		
0008 70A1h	ICU	Interrupt request register 161	IR161	8	8	2 ICLK		
0008 70A2h	ICU	Interrupt request register 162	IR162	8	8	2 ICLK		
0008 70A3h	ICU	Interrupt request register 163	IR163	8	8	2 ICLK		
0008 70A4h	ICU	Interrupt request register 164	IR164	8	8	2 ICLK		
0008 70A5h	ICU	Interrupt request register 165	IR165	8	8	2 ICLK		
0008 70A6h	ICU	Interrupt request register 166	IR166	8	8	2 ICLK		
0008 70A7h	ICU	Interrupt request register 167	IR167	8	8	2 ICLK		
0008 70AAh	ICU	Interrupt request register 170	IR170	8	8	2 ICLK		
0008 70ABh	ICU	Interrupt request register 171	IR171	8	8	2 ICLK		
0008 70ACh	ICU	Interrupt request register 172	IR172	8	8	2 ICLK		
0008 70ADh	ICU	Interrupt request register 173	IR173	8	8	2 ICLK		
0008 70AEh	ICU	Interrupt request register 174	IR174	8	8	2 ICLK		
0008 70AFh	ICU	Interrupt request register 175	IR175	8	8	2 ICLK		
0008 70B0h	ICU	Interrupt request register 176	IR176	8	8	2 ICLK		
0008 70B1h	ICU	Interrupt request register 177	IR177	8	8	2 ICLK		
0008 70B2h	ICU	Interrupt request register 178	IR178	8	8	2 ICLK		
0008 70B3h	ICU	Interrupt request register 179	IR179	8	8	2 ICLK		
0008 70B4h	ICU	Interrupt request register 180	IR180	8	8	2 ICLK		
0008 70B5h	ICU	Interrupt request register 181	IR181	8	8	2 ICLK		
0008 70B6h	ICU	Interrupt request register 182	IR182	8	8	2 ICLK		
0008 70B7h	ICU	Interrupt request register 183	IR183	8	8	2 ICLK		
0008 70B8h	ICU	Interrupt request register 184	IR184	8	8	2 ICLK		
0008 70B9h	ICU	Interrupt request register 185	IR185	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (7/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 70BAh	ICU	Interrupt request register 186	IR186	8	8	2 ICLK		section 15.
0008 70BBh	ICU	Interrupt request register 187	IR187	8	8	2 ICLK		
0008 70BCh	ICU	Interrupt request register 188	IR188	8	8	2 ICLK		
0008 70BDh	ICU	Interrupt request register 189	IR189	8	8	2 ICLK		
0008 70BEh	ICU	Interrupt request register 190	IR190	8	8	2 ICLK		
0008 70BFh	ICU	Interrupt request register 191	IR191	8	8	2 ICLK		
0008 70C0h	ICU	Interrupt request register 192	IR192	8	8	2 ICLK		
0008 70C1h	ICU	Interrupt request register 193	IR193	8	8	2 ICLK		
0008 70C2h	ICU	Interrupt request register 194	IR194	8	8	2 ICLK		
0008 70C3h	ICU	Interrupt request register 195	IR195	8	8	2 ICLK		
0008 70C4h	ICU	Interrupt request register 196	IR196	8	8	2 ICLK		
0008 70C5h	ICU	Interrupt request register 197	IR197	8	8	2 ICLK		
0008 70C6h	ICU	Interrupt request register 198	IR198	8	8	2 ICLK		
0008 70C7h	ICU	Interrupt request register 199	IR199	8	8	2 ICLK		
0008 70C8h	ICU	Interrupt request register 200	IR200	8	8	2 ICLK		
0008 70C9h	ICU	Interrupt request register 201	IR201	8	8	2 ICLK		
0008 70CAh	ICU	Interrupt request register 202	IR202	8	8	2 ICLK		
0008 70CBh	ICU	Interrupt request register 203	IR203	8	8	2 ICLK		
0008 70D6h	ICU	Interrupt request register 214	IR214	8	8	2 ICLK		
0008 70D7h	ICU	Interrupt request register 215	IR215	8	8	2 ICLK		
0008 70D8h	ICU	Interrupt request register 216	IR216	8	8	2 ICLK		
0008 70D9h	ICU	Interrupt request register 217	IR217	8	8	2 ICLK		
0008 70DAh	ICU	Interrupt request register 218	IR218	8	8	2 ICLK		
0008 70DBh	ICU	Interrupt request register 219	IR219	8	8	2 ICLK		
0008 70DCh	ICU	Interrupt request register 220	IR220	8	8	2 ICLK		
0008 70DDh	ICU	Interrupt request register 221	IR221	8	8	2 ICLK		
0008 70DEh	ICU	Interrupt request register 222	IR222	8	8	2 ICLK		
0008 70DFh	ICU	Interrupt request register 223	IR223	8	8	2 ICLK		
0008 70E0h	ICU	Interrupt request register 224	IR224	8	8	2 ICLK		
0008 70E1h	ICU	Interrupt request register 225	IR225	8	8	2 ICLK		
0008 70E2h	ICU	Interrupt request register 226	IR226	8	8	2 ICLK		
0008 70E3h	ICU	Interrupt request register 227	IR227	8	8	2 ICLK		
0008 70E4h	ICU	Interrupt request register 228	IR228	8	8	2 ICLK		
0008 70E5h	ICU	Interrupt request register 229	IR229	8	8	2 ICLK		
0008 70E6h	ICU	Interrupt request register 230	IR230	8	8	2 ICLK		
0008 70E7h	ICU	Interrupt request register 231	IR231	8	8	2 ICLK		
0008 70E8h	ICU	Interrupt request register 232	IR232	8	8	2 ICLK		
0008 70E9h	ICU	Interrupt request register 233	IR233	8	8	2 ICLK		
0008 70EAh	ICU	Interrupt request register 234	IR234	8	8	2 ICLK		
0008 70EBh	ICU	Interrupt request register 235	IR235	8	8	2 ICLK		
0008 70ECh	ICU	Interrupt request register 236	IR236	8	8	2 ICLK		
0008 70EDh	ICU	Interrupt request register 237	IR237	8	8	2 ICLK		
0008 70EEh	ICU	Interrupt request register 238	IR238	8	8	2 ICLK		
0008 70EFh	ICU	Interrupt request register 239	IR239	8	8	2 ICLK		
0008 70F0h	ICU	Interrupt request register 240	IR240	8	8	2 ICLK		
0008 70F1h	ICU	Interrupt request register 241	IR241	8	8	2 ICLK		
0008 70F2h	ICU	Interrupt request register 242	IR242	8	8	2 ICLK		
0008 70F3h	ICU	Interrupt request register 243	IR243	8	8	2 ICLK		
0008 70F4h	ICU	Interrupt request register 244	IR244	8	8	2 ICLK		
0008 70F5h	ICU	Interrupt request register 245	IR245	8	8	2 ICLK		
0008 70F6h	ICU	Interrupt request register 246	IR246	8	8	2 ICLK		
0008 70F7h	ICU	Interrupt request register 247	IR247	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (8/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 70F8h	ICU	Interrupt request register 248	IR248	8	8	2 ICLK		section 15.
0008 70F9h	ICU	Interrupt request register 249	IR249	8	8	2 ICLK		
0008 70FAh	ICU	Interrupt request register 250	IR250	8	8	2 ICLK		
0008 70FBh	ICU	Interrupt request register 251	IR251	8	8	2 ICLK		
0008 70FCh	ICU	Interrupt request register 252	IR252	8	8	2 ICLK		
0008 70FDh	ICU	Interrupt request register 253	IR253	8	8	2 ICLK		
0008 711Bh	ICU	DTC activation enable register 027	DTCER027	8	8	2 ICLK		
0008 711Ch	ICU	DTC activation enable register 028	DTCER028	8	8	2 ICLK		
0008 711Dh	ICU	DTC activation enable register 029	DTCER029	8	8	2 ICLK		
0008 711Eh	ICU	DTC activation enable register 030	DTCER030	8	8	2 ICLK		
0008 711Fh	ICU	DTC activation enable register 031	DTCER031	8	8	2 ICLK		
0008 7121h	ICU	DTC activation enable register 033	DTCER033	8	8	2 ICLK		
0008 7122h	ICU	DTC activation enable register 034	DTCER034	8	8	2 ICLK		
0008 7124h	ICU	DTC activation enable register 036	DTCER036	8	8	2 ICLK		
0008 7125h	ICU	DTC activation enable register 037	DTCER037	8	8	2 ICLK		
0008 7127h	ICU	DTC activation enable register 039	DTCER039	8	8	2 ICLK		
0008 7128h	ICU	DTC activation enable register 040	DTCER040	8	8	2 ICLK		
0008 712Ah	ICU	DTC activation enable register 042	DTCER042	8	8	2 ICLK		
0008 712Bh	ICU	DTC activation enable register 043	DTCER043	8	8	2 ICLK		
0008 712Dh	ICU	DTC activation enable register 045	DTCER045	8	8	2 ICLK		
0008 712Eh	ICU	DTC activation enable register 046	DTCER046	8	8	2 ICLK		
0008 7140h	ICU	DTC activation enable register 064	DTCER064	8	8	2 ICLK		
0008 7141h	ICU	DTC activation enable register 065	DTCER065	8	8	2 ICLK		
0008 7142h	ICU	DTC activation enable register 066	DTCER066	8	8	2 ICLK		
0008 7143h	ICU	DTC activation enable register 067	DTCER067	8	8	2 ICLK		
0008 7144h	ICU	DTC activation enable register 068	DTCER068	8	8	2 ICLK		
0008 7145h	ICU	DTC activation enable register 069	DTCER069	8	8	2 ICLK		
0008 7146h	ICU	DTC activation enable register 070	DTCER070	8	8	2 ICLK		
0008 7147h	ICU	DTC activation enable register 071	DTCER071	8	8	2 ICLK		
0008 7148h	ICU	DTC activation enable register 072	DTCER072	8	8	2 ICLK		
0008 7149h	ICU	DTC activation enable register 073	DTCER073	8	8	2 ICLK		
0008 714Ah	ICU	DTC activation enable register 074	DTCER074	8	8	2 ICLK		
0008 714Bh	ICU	DTC activation enable register 075	DTCER075	8	8	2 ICLK		
0008 714Ch	ICU	DTC activation enable register 076	DTCER076	8	8	2 ICLK		
0008 714Dh	ICU	DTC activation enable register 077	DTCER077	8	8	2 ICLK		
0008 714Eh	ICU	DTC activation enable register 078	DTCER078	8	8	2 ICLK		
0008 714Fh	ICU	DTC activation enable register 079	DTCER079	8	8	2 ICLK		
0008 7162h	ICU	DTC activation enable register 098	DTCER098	8	8	2 ICLK		
0008 7166h	ICU	DTC activation enable register 102	DTCER102	8	8	2 ICLK		
0008 717Eh	ICU	DTC activation enable register 126	DTCER126	8	8	2 ICLK		
0008 717Fh	ICU	DTC activation enable register 127	DTCER127	8	8	2 ICLK		
0008 7180h	ICU	DTC activation enable register 128	DTCER128	8	8	2 ICLK		
0008 7181h	ICU	DTC activation enable register 129	DTCER129	8	8	2 ICLK		
0008 7182h	ICU	DTC activation enable register 130	DTCER130	8	8	2 ICLK		
0008 7183h	ICU	DTC activation enable register 131	DTCER131	8	8	2 ICLK		
0008 7184h	ICU	DTC activation enable register 132	DTCER132	8	8	2 ICLK		
0008 7185h	ICU	DTC activation enable register 133	DTCER133	8	8	2 ICLK		
0008 7186h	ICU	DTC activation enable register 134	DTCER134	8	8	2 ICLK		
0008 7187h	ICU	DTC activation enable register 135	DTCER135	8	8	2 ICLK		
0008 7188h	ICU	DTC activation enable register 136	DTCER136	8	8	2 ICLK		
0008 7189h	ICU	DTC activation enable register 137	DTCER137	8	8	2 ICLK		
0008 718Ah	ICU	DTC activation enable register 138	DTCER138	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (9/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 718Bh	ICU	DTC activation enable register 139	DTCER139	8	8	2 ICLK		section 15.
0008 718Ch	ICU	DTC activation enable register 140	DTCER140	8	8	2 ICLK		
0008 718Dh	ICU	DTC activation enable register 141	DTCER141	8	8	2 ICLK		
0008 718Eh	ICU	DTC activation enable register 142	DTCER142	8	8	2 ICLK		
0008 718Fh	ICU	DTC activation enable register 143	DTCER143	8	8	2 ICLK		
0008 7190h	ICU	DTC activation enable register 144	DTCER144	8	8	2 ICLK		
0008 7191h	ICU	DTC activation enable register 145	DTCER145	8	8	2 ICLK		
0008 7194h	ICU	DTC activation enable register 148	DTCER148	8	8	2 ICLK		
0008 7195h	ICU	DTC activation enable register 149	DTCER149	8	8	2 ICLK		
0008 7196h	ICU	DTC activation enable register 150	DTCER150	8	8	2 ICLK		
0008 7197h	ICU	DTC activation enable register 151	DTCER151	8	8	2 ICLK		
0008 7198h	ICU	DTC activation enable register 152	DTCER152	8	8	2 ICLK		
0008 7199h	ICU	DTC activation enable register 153	DTCER153	8	8	2 ICLK		
0008 719Ah	ICU	DTC activation enable register 154	DTCER154	8	8	2 ICLK		
0008 719Bh	ICU	DTC activation enable register 155	DTCER155	8	8	2 ICLK		
0008 719Ch	ICU	DTC activation enable register 156	DTCER156	8	8	2 ICLK		
0008 719Dh	ICU	DTC activation enable register 157	DTCER157	8	8	2 ICLK		
0008 719Eh	ICU	DTC activation enable register 158	DTCER158	8	8	2 ICLK		
0008 719Fh	ICU	DTC activation enable register 159	DTCER159	8	8	2 ICLK		
0008 71A0h	ICU	DTC activation enable register 160	DTCER160	8	8	2 ICLK		
0008 71A1h	ICU	DTC activation enable register 161	DTCER161	8	8	2 ICLK		
0008 71A2h	ICU	DTC activation enable register 162	DTCER162	8	8	2 ICLK		
0008 71A3h	ICU	DTC activation enable register 163	DTCER163	8	8	2 ICLK		
0008 71A4h	ICU	DTC activation enable register 164	DTCER164	8	8	2 ICLK		
0008 71A5h	ICU	DTC activation enable register 165	DTCER165	8	8	2 ICLK		
0008 71AAh	ICU	DTC activation enable register 170	DTCER170	8	8	2 ICLK		
0008 71ABh	ICU	DTC activation enable register 171	DTCER171	8	8	2 ICLK		
0008 71ADh	ICU	DTC activation enable register 173	DTCER173	8	8	2 ICLK		
0008 71AEh	ICU	DTC activation enable register 174	DTCER174	8	8	2 ICLK		
0008 71B0h	ICU	DTC activation enable register 176	DTCER176	8	8	2 ICLK		
0008 71B1h	ICU	DTC activation enable register 177	DTCER177	8	8	2 ICLK		
0008 71B3h	ICU	DTC activation enable register 179	DTCER179	8	8	2 ICLK		
0008 71B4h	ICU	DTC activation enable register 180	DTCER180	8	8	2 ICLK		
0008 71B7h	ICU	DTC activation enable register 183	DTCER183	8	8	2 ICLK		
0008 71B8h	ICU	DTC activation enable register 184	DTCER184	8	8	2 ICLK		
0008 71BBh	ICU	DTC activation enable register 187	DTCER187	8	8	2 ICLK		
0008 71BCh	ICU	DTC activation enable register 188	DTCER188	8	8	2 ICLK		
0008 71BFh	ICU	DTC activation enable register 191	DTCER191	8	8	2 ICLK		
0008 71C0h	ICU	DTC activation enable register 192	DTCER192	8	8	2 ICLK		
0008 71C3h	ICU	DTC activation enable register 195	DTCER195	8	8	2 ICLK		
0008 71C4h	ICU	DTC activation enable register 196	DTCER196	8	8	2 ICLK		
0008 71C6h	ICU	DTC activation enable register 198	DTCER198	8	8	2 ICLK		
0008 71C7h	ICU	DTC activation enable register 199	DTCER199	8	8	2 ICLK		
0008 71C8h	ICU	DTC activation enable register 200	DTCER200	8	8	2 ICLK		
0008 71C9h	ICU	DTC activation enable register 201	DTCER201	8	8	2 ICLK		
0008 71CAh	ICU	DTC activation enable register 202	DTCER202	8	8	2 ICLK		
0008 71CBh	ICU	DTC activation enable register 203	DTCER203	8	8	2 ICLK		
0008 71D6h	ICU	DTC activation enable register 214	DTCER214	8	8	2 ICLK		
0008 71D7h	ICU	DTC activation enable register 215	DTCER215	8	8	2 ICLK		
0008 71D9h	ICU	DTC activation enable register 217	DTCER217	8	8	2 ICLK		
0008 71DAh	ICU	DTC activation enable register 218	DTCER218	8	8	2 ICLK		
0008 71DCh	ICU	DTC activation enable register 220	DTCER220	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (10/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 71DDh	ICU	DTC activation enable register 221	DTCER221	8	8	2 ICLK		section 15.
0008 71DFh	ICU	DTC activation enable register 223	DTCER223	8	8	2 ICLK		
0008 71E0h	ICU	DTC activation enable register 224	DTCER224	8	8	2 ICLK		
0008 71E2h	ICU	DTC activation enable register 226	DTCER226	8	8	2 ICLK		
0008 71E3h	ICU	DTC activation enable register 227	DTCER227	8	8	2 ICLK		
0008 71E5h	ICU	DTC activation enable register 229	DTCER229	8	8	2 ICLK		
0008 71E6h	ICU	DTC activation enable register 230	DTCER230	8	8	2 ICLK		
0008 71E8h	ICU	DTC activation enable register 232	DTCER232	8	8	2 ICLK		
0008 71E9h	ICU	DTC activation enable register 233	DTCER233	8	8	2 ICLK		
0008 71EBh	ICU	DTC activation enable register 235	DTCER235	8	8	2 ICLK		
0008 71ECh	ICU	DTC activation enable register 236	DTCER236	8	8	2 ICLK		
0008 71EEh	ICU	DTC activation enable register 238	DTCER238	8	8	2 ICLK		
0008 71EFh	ICU	DTC activation enable register 239	DTCER239	8	8	2 ICLK		
0008 71F1h	ICU	DTC activation enable register 241	DTCER241	8	8	2 ICLK		
0008 71F2h	ICU	DTC activation enable register 242	DTCER242	8	8	2 ICLK		
0008 71F4h	ICU	DTC activation enable register 244	DTCER244	8	8	2 ICLK		
0008 71F5h	ICU	DTC activation enable register 245	DTCER245	8	8	2 ICLK		
0008 71F7h	ICU	DTC activation enable register 247	DTCER247	8	8	2 ICLK		
0008 71F8h	ICU	DTC activation enable register 248	DTCER248	8	8	2 ICLK		
0008 71FAh	ICU	DTC activation enable register 250	DTCER250	8	8	2 ICLK		
0008 71FBh	ICU	DTC activation enable register 251	DTCER251	8	8	2 ICLK		
0008 7202h	ICU	Interrupt request enable register 02	IER02	8	8	2 ICLK		
0008 7203h	ICU	Interrupt request enable register 03	IER03	8	8	2 ICLK		
0008 7204h	ICU	Interrupt request enable register 04	IER04	8	8	2 ICLK		
0008 7205h	ICU	Interrupt request enable register 05	IER05	8	8	2 ICLK		
0008 7206h	ICU	Interrupt request enable register 06	IER06	8	8	2 ICLK		
0008 7207h	ICU	Interrupt request enable register 07	IER07	8	8	2 ICLK		
0008 7208h	ICU	Interrupt request enable register 08	IER08	8	8	2 ICLK		
0008 7209h	ICU	Interrupt request enable register 09	IER09	8	8	2 ICLK		
0008 720Bh	ICU	Interrupt request enable register 0B	IER0B	8	8	2 ICLK		
0008 720Ch	ICU	Interrupt request enable register 0C	IER0C	8	8	2 ICLK		
0008 720Dh	ICU	Interrupt request enable register 0D	IER0D	8	8	2 ICLK		
0008 720Eh	ICU	Interrupt request enable register 0E	IER0E	8	8	2 ICLK		
0008 720Fh	ICU	Interrupt request enable register 0F	IER0F	8	8	2 ICLK		
0008 7210h	ICU	Interrupt request enable register 10	IER10	8	8	2 ICLK		
0008 7211h	ICU	Interrupt request enable register 11	IER11	8	8	2 ICLK		
0008 7212h	ICU	Interrupt request enable register 12	IER12	8	8	2 ICLK		
0008 7213h	ICU	Interrupt request enable register 13	IER13	8	8	2 ICLK		
0008 7214h	ICU	Interrupt request enable register 14	IER14	8	8	2 ICLK		
0008 7215h	ICU	Interrupt request enable register 15	IER15	8	8	2 ICLK		
0008 7216h	ICU	Interrupt request enable register 16	IER16	8	8	2 ICLK		
0008 7217h	ICU	Interrupt request enable register 17	IER17	8	8	2 ICLK		
0008 7218h	ICU	Interrupt request enable register 18	IER18	8	8	2 ICLK		
0008 7219h	ICU	Interrupt request enable register 19	IER19	8	8	2 ICLK		
0008 721Ah	ICU	Interrupt request enable register 1A	IER1A	8	8	2 ICLK		
0008 721Bh	ICU	Interrupt request enable register 1B	IER1B	8	8	2 ICLK		
0008 721Ch	ICU	Interrupt request enable register 1C	IER1C	8	8	2 ICLK		
0008 721Dh	ICU	Interrupt request enable register 1D	IER1D	8	8	2 ICLK		
0008 721Eh	ICU	Interrupt request enable register 1E	IER1E	8	8	2 ICLK		
0008 721Fh	ICU	Interrupt request enable register 1F	IER1F	8	8	2 ICLK		
0008 72E0h	ICU	Software interrupt activation register	SWINTR	8	8	2 ICLK		
0008 72F0h	ICU	Fast interrupt register	FIR	16	16	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (11/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 7300h	ICU	Interrupt source priority register 000	IPR000	8	8	2 ICLK		section 15.
0008 7301h	ICU	Interrupt source priority register 001	IPR001	8	8	2 ICLK		
0008 7302h	ICU	Interrupt source priority register 002	IPR002	8	8	2 ICLK		
0008 7303h	ICU	Interrupt source priority register 003	IPR003	8	8	2 ICLK		
0008 7304h	ICU	Interrupt source priority register 004	IPR004	8	8	2 ICLK		
0008 7305h	ICU	Interrupt source priority register 005	IPR005	8	8	2 ICLK		
0008 7306h	ICU	Interrupt source priority register 006	IPR006	8	8	2 ICLK		
0008 7307h	ICU	Interrupt source priority register 007	IPR007	8	8	2 ICLK		
0008 7320h	ICU	Interrupt source priority register 032	IPR032	8	8	2 ICLK		
0008 7321h	ICU	Interrupt source priority register 033	IPR033	8	8	2 ICLK		
0008 7322h	ICU	Interrupt source priority register 034	IPR034	8	8	2 ICLK		
0008 7323h	ICU	Interrupt source priority register 035	IPR035	8	8	2 ICLK		
0008 7324h	ICU	Interrupt source priority register 036	IPR036	8	8	2 ICLK		
0008 7325h	ICU	Interrupt source priority register 037	IPR037	8	8	2 ICLK		
0008 7326h	ICU	Interrupt source priority register 038	IPR038	8	8	2 ICLK		
0008 7327h	ICU	Interrupt source priority register 039	IPR039	8	8	2 ICLK		
0008 732Ah	ICU	Interrupt source priority register 042	IPR042	8	8	2 ICLK		
0008 732Dh	ICU	Interrupt source priority register 045	IPR045	8	8	2 ICLK		
0008 7330h	ICU	Interrupt source priority register 048	IPR048	8	8	2 ICLK		
0008 7334h	ICU	Interrupt source priority register 052	IPR052	8	8	2 ICLK		
0008 7338h	ICU	Interrupt source priority register 056	IPR056	8	8	2 ICLK		
0008 733Eh	ICU	Interrupt source priority register 062	IPR062	8	8	2 ICLK		
0008 7340h	ICU	Interrupt source priority register 064	IPR064	8	8	2 ICLK		
0008 7341h	ICU	Interrupt source priority register 065	IPR065	8	8	2 ICLK		
0008 7342h	ICU	Interrupt source priority register 066	IPR066	8	8	2 ICLK		
0008 7343h	ICU	Interrupt source priority register 067	IPR067	8	8	2 ICLK		
0008 7344h	ICU	Interrupt source priority register 068	IPR068	8	8	2 ICLK		
0008 7345h	ICU	Interrupt source priority register 069	IPR069	8	8	2 ICLK		
0008 7346h	ICU	Interrupt source priority register 070	IPR070	8	8	2 ICLK		
0008 7347h	ICU	Interrupt source priority register 071	IPR071	8	8	2 ICLK		
0008 7348h	ICU	Interrupt source priority register 072	IPR072	8	8	2 ICLK		
0008 7349h	ICU	Interrupt source priority register 073	IPR073	8	8	2 ICLK		
0008 734Ah	ICU	Interrupt source priority register 074	IPR074	8	8	2 ICLK		
0008 734Bh	ICU	Interrupt source priority register 075	IPR075	8	8	2 ICLK		
0008 734Ch	ICU	Interrupt source priority register 076	IPR076	8	8	2 ICLK		
0008 734Dh	ICU	Interrupt source priority register 077	IPR077	8	8	2 ICLK		
0008 734Eh	ICU	Interrupt source priority register 078	IPR078	8	8	2 ICLK		
0008 734Fh	ICU	Interrupt source priority register 079	IPR079	8	8	2 ICLK		
0008 735Ah	ICU	Interrupt source priority register 090	IPR090	8	8	2 ICLK		
0008 735Bh	ICU	Interrupt source priority register 091	IPR091	8	8	2 ICLK		
0008 735Ch	ICU	Interrupt source priority register 092	IPR092	8	8	2 ICLK		
0008 735Dh	ICU	Interrupt source priority register 093	IPR093	8	8	2 ICLK		
0008 7362h	ICU	Interrupt source priority register 098	IPR098	8	8	2 ICLK		
0008 7366h	ICU	Interrupt source priority register 102	IPR102	8	8	2 ICLK		
0008 736Ah	ICU	Interrupt source priority register 106	IPR106	8	8	2 ICLK		
0008 736Bh	ICU	Interrupt source priority register 107	IPR107	8	8	2 ICLK		
0008 736Ch	ICU	Interrupt source priority register 108	IPR108	8	8	2 ICLK		
0008 736Dh	ICU	Interrupt source priority register 109	IPR109	8	8	2 ICLK		
0008 736Eh	ICU	Interrupt source priority register 110	IPR110	8	8	2 ICLK		
0008 736Fh	ICU	Interrupt source priority register 111	IPR111	8	8	2 ICLK		
0008 7370h	ICU	Interrupt source priority register 112	IPR112	8	8	2 ICLK		
0008 7372h	ICU	Interrupt source priority register 114	IPR114	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (12/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 737Ah	ICU	Interrupt source priority register 122	IPR122	8	8	2 ICLK		section 15.
0008 737Eh	ICU	Interrupt source priority register 126	IPR126	8	8	2 ICLK		
0008 7382h	ICU	Interrupt source priority register 130	IPR130	8	8	2 ICLK		
0008 7384h	ICU	Interrupt source priority register 132	IPR132	8	8	2 ICLK		
0008 7386h	ICU	Interrupt source priority register 134	IPR134	8	8	2 ICLK		
0008 738Ah	ICU	Interrupt source priority register 138	IPR138	8	8	2 ICLK		
0008 738Ch	ICU	Interrupt source priority register 140	IPR140	8	8	2 ICLK		
0008 738Eh	ICU	Interrupt source priority register 142	IPR142	8	8	2 ICLK		
0008 7392h	ICU	Interrupt source priority register 146	IPR146	8	8	2 ICLK		
0008 7394h	ICU	Interrupt source priority register 148	IPR148	8	8	2 ICLK		
0008 7396h	ICU	Interrupt source priority register 150	IPR150	8	8	2 ICLK		
0008 7398h	ICU	Interrupt source priority register 152	IPR152	8	8	2 ICLK		
0008 739Ch	ICU	Interrupt source priority register 156	IPR156	8	8	2 ICLK		
0008 73A0h	ICU	Interrupt source priority register 160	IPR160	8	8	2 ICLK		
0008 73A1h	ICU	Interrupt source priority register 161	IPR161	8	8	2 ICLK		
0008 73A4h	ICU	Interrupt source priority register 164	IPR164	8	8	2 ICLK		
0008 73A6h	ICU	Interrupt source priority register 166	IPR166	8	8	2 ICLK		
0008 73AAh	ICU	Interrupt source priority register 170	IPR170	8	8	2 ICLK		
0008 73ADh	ICU	Interrupt source priority register 173	IPR173	8	8	2 ICLK		
0008 73B0h	ICU	Interrupt source priority register 176	IPR176	8	8	2 ICLK		
0008 73B3h	ICU	Interrupt source priority register 179	IPR179	8	8	2 ICLK		
0008 73B6h	ICU	Interrupt source priority register 182	IPR182	8	8	2 ICLK		
0008 73B7h	ICU	Interrupt source priority register 183	IPR183	8	8	2 ICLK		
0008 73B8h	ICU	Interrupt source priority register 184	IPR184	8	8	2 ICLK		
0008 73B9h	ICU	Interrupt source priority register 185	IPR185	8	8	2 ICLK		
0008 73BAh	ICU	Interrupt source priority register 186	IPR186	8	8	2 ICLK		
0008 73BBh	ICU	Interrupt source priority register 187	IPR187	8	8	2 ICLK		
0008 73BCh	ICU	Interrupt source priority register 188	IPR188	8	8	2 ICLK		
0008 73BDh	ICU	Interrupt source priority register 189	IPR189	8	8	2 ICLK		
0008 73BEh	ICU	Interrupt source priority register 190	IPR190	8	8	2 ICLK		
0008 73BFh	ICU	Interrupt source priority register 191	IPR191	8	8	2 ICLK		
0008 73C0h	ICU	Interrupt source priority register 192	IPR192	8	8	2 ICLK		
0008 73C1h	ICU	Interrupt source priority register 193	IPR193	8	8	2 ICLK		
0008 73C2h	ICU	Interrupt source priority register 194	IPR194	8	8	2 ICLK		
0008 73C3h	ICU	Interrupt source priority register 195	IPR195	8	8	2 ICLK		
0008 73C4h	ICU	Interrupt source priority register 196	IPR196	8	8	2 ICLK		
0008 73C5h	ICU	Interrupt source priority register 197	IPR197	8	8	2 ICLK		
0008 73C6h	ICU	Interrupt source priority register 198	IPR198	8	8	2 ICLK		
0008 73C7h	ICU	Interrupt source priority register 199	IPR199	8	8	2 ICLK		
0008 73C8h	ICU	Interrupt source priority register 200	IPR200	8	8	2 ICLK		
0008 73C9h	ICU	Interrupt source priority register 201	IPR201	8	8	2 ICLK		
0008 73CAh	ICU	Interrupt source priority register 202	IPR202	8	8	2 ICLK		
0008 73CBh	ICU	Interrupt source priority register 203	IPR203	8	8	2 ICLK		
0008 73D6h	ICU	Interrupt source priority register 214	IPR214	8	8	2 ICLK		
0008 73D9h	ICU	Interrupt source priority register 217	IPR217	8	8	2 ICLK		
0008 73DCh	ICU	Interrupt source priority register 220	IPR220	8	8	2 ICLK		
0008 73DFh	ICU	Interrupt source priority register 223	IPR223	8	8	2 ICLK		
0008 73E2h	ICU	Interrupt source priority register 226	IPR226	8	8	2 ICLK		
0008 73E5h	ICU	Interrupt source priority register 229	IPR229	8	8	2 ICLK		
0008 73E8h	ICU	Interrupt source priority register 232	IPR232	8	8	2 ICLK		
0008 73EBh	ICU	Interrupt source priority register 235	IPR235	8	8	2 ICLK		
0008 73EEh	ICU	Interrupt source priority register 238	IPR238	8	8	2 ICLK		

Table 5.1 List of I/O Registers (Address Order) (13/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 73F1h	ICU	Interrupt source priority register 241	IPR241	8	8	2 ICLK		section 15.
0008 73F4h	ICU	Interrupt source priority register 244	IPR244	8	8	2 ICLK		
0008 73F7h	ICU	Interrupt source priority register 247	IPR247	8	8	2 ICLK		
0008 73FAh	ICU	Interrupt source priority register 250	IPR250	8	8	2 ICLK		
0008 73FDh	ICU	Interrupt source priority register 253	IPR253	8	8	2 ICLK		
0008 7400h	ICU	DMAC activation source select register 0	DMRSR0	8	8	2 ICLK		
0008 7404h	ICU	DMAC activation source select register 1	DMRSR1	8	8	2 ICLK		
0008 7408h	ICU	DMAC activation source select register 2	DMRSR2	8	8	2 ICLK		
0008 740Ch	ICU	DMAC activation source select register 3	DMRSR3	8	8	2 ICLK		
0008 7500h	ICU	IRQ control register 0	IRQCR0	8	8	2 ICLK		
0008 7501h	ICU	IRQ control register 1	IRQCR1	8	8	2 ICLK		
0008 7502h	ICU	IRQ control register 2	IRQCR2	8	8	2 ICLK		
0008 7503h	ICU	IRQ control register 3	IRQCR3	8	8	2 ICLK		
0008 7504h	ICU	IRQ control register 4	IRQCR4	8	8	2 ICLK		
0008 7505h	ICU	IRQ control register 5	IRQCR5	8	8	2 ICLK		
0008 7506h	ICU	IRQ control register 6	IRQCR6	8	8	2 ICLK		
0008 7507h	ICU	IRQ control register 7	IRQCR7	8	8	2 ICLK		
0008 7508h	ICU	IRQ control register 8	IRQCR8	8	8	2 ICLK		
0008 7509h	ICU	IRQ control register 9	IRQCR9	8	8	2 ICLK		
0008 750Ah	ICU	IRQ control register 10	IRQCR10	8	8	2 ICLK		
0008 750Bh	ICU	IRQ control register 11	IRQCR11	8	8	2 ICLK		
0008 750Ch	ICU	IRQ control register 12	IRQCR12	8	8	2 ICLK		
0008 750Dh	ICU	IRQ control register 13	IRQCR13	8	8	2 ICLK		
0008 750Eh	ICU	IRQ control register 14	IRQCR14	8	8	2 ICLK		
0008 750Fh	ICU	IRQ control register 15	IRQCR15	8	8	2 ICLK		
0008 7510h	ICU	IRQ pin digital filter enable register 0	IRQFLTE0	8	8	2 ICLK		
0008 7511h	ICU	IRQ pin digital filter enable register 1	IRQFLTE1	8	8	2 ICLK		
0008 7514h	ICU	IRQ pin digital filter enable register 0	IRQFLTC0	16	16	2 ICLK		
0008 7516h	ICU	IRQ pin digital filter enable register 1	IRQFLTC1	16	16	2 ICLK		
0008 7580h	ICU	Non-maskable interrupt status register	NMISR	8	8	2 ICLK		
0008 7581h	ICU	Non-maskable interrupt enable register	NMIER	8	8	2 ICLK		
0008 7582h	ICU	Non-maskable interrupt clear register	NMICLR	8	8	2 ICLK		
0008 7583h	ICU	NMI pin interrupt control register	NMICR	8	8	2 ICLK		
0008 7590h	ICU	NMI pin digital filter enable register	NMIFLTE	8	8	2 ICLK		
0008 7594h	ICU	NMI pin digital filter setting register	NMIFLTC	16	16	2 ICLK		
0008 8000h	CMT	Compare match timer start register 0	CMSTR0	16	16	2, 3 PCLKB	2 ICLK	section 27.
0008 8002h	CMT0	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK	
0008 8004h	CMT0	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8006h	CMT0	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	
0008 8008h	CMT1	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK	
0008 800Ah	CMT1	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 800Ch	CMT1	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	
0008 8010h	CMT	Compare match timer start register 1	CMSTR1	16	16	2, 3 PCLKB	2 ICLK	
0008 8012h	CMT2	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK	
0008 8014h	CMT2	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8016h	CMT2	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	
0008 8018h	CMT3	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK	
0008 801Ah	CMT3	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 801Ch	CMT3	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	
0008 8020h	WDT	WDT refresh register	WDTRR	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8022h	WDT	WDT control register	WDTCR	16	16	2, 3 PCLKB	2 ICLK	
0008 8024h	WDT	WDT status register	WDTSR	16	16	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (14/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 8026h	WDT	WDT reset control register	WDTRCR	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8030h	IWDT	IWDT refresh register	IWDTRR	8	8	2, 3 PCLKB	2 ICLK	
0008 8032h	IWDT	IWDT control register	IWDTCR	16	16	2, 3 PCLKB	2 ICLK	
0008 8034h	IWDT	IWDT status register	IWDTSR	16	16	2, 3 PCLKB	2 ICLK	
0008 8036h	IWDT	IWDT reset control register	IWDTRCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8038h	IWDT	IWDT count stop control register	IWDTCSTPR	8	8	2, 3 PCLKB	2 ICLK	
0008 80C0h	DA	D/A data register 0	DADR0	16	16	2, 3 PCLKB	2 ICLK	section 42.
0008 80C2h	DA	D/A data register 1	DADR1	16	16	2, 3 PCLKB	2 ICLK	
0008 80C4h	DA	D/A control register	DACR	8	8	2, 3 PCLKB	2 ICLK	
0008 80C5h	DA	DADRm format select register	DADPR	8	8	2, 3 PCLKB	2 ICLK	
0008 80C6h	DA	D/A A/D synchronous start control register	DAADSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8100h	TPUA	Timer start register	TSTR	8	8	2, 3 PCLKB	2 ICLK	section 24.
0008 8101h	TPUA	Timer synchronous register	TSYR	8	8	2, 3 PCLKB	2 ICLK	
0008 8108h	TPU0	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8109h	TPU1	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 810Ah	TPU2	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 810Bh	TPU3	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 810Ch	TPU4	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 810Dh	TPU5	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8110h	TPU0	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8111h	TPU0	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8112h	TPU0	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	
0008 8113h	TPU0	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	
0008 8114h	TPU0	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8115h	TPU0	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8116h	TPU0	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8118h	TPU0	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 811Ah	TPU0	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 811Ch	TPU0	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	
0008 811Eh	TPU0	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	
0008 8120h	TPU1	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8121h	TPU1	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8122h	TPU1	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 8124h	TPU1	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8125h	TPU1	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8126h	TPU1	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8128h	TPU1	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 812Ah	TPU1	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 8130h	TPU2	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8131h	TPU2	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8132h	TPU2	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 8134h	TPU2	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8135h	TPU2	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8136h	TPU2	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8138h	TPU2	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 813Ah	TPU2	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 8140h	TPU3	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8141h	TPU3	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8142h	TPU3	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	
0008 8143h	TPU3	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	
0008 8144h	TPU3	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8145h	TPU3	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (15/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 8146h	TPU3	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8148h	TPU3	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 814Ah	TPU3	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 814Ch	TPU3	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	
0008 814Eh	TPU3	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	
0008 8150h	TPU4	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8151h	TPU4	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8152h	TPU4	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 8154h	TPU4	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8155h	TPU4	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8156h	TPU4	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8158h	TPU4	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 815Ah	TPU4	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 8160h	TPU5	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8161h	TPU5	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8162h	TPU5	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 8164h	TPU5	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8165h	TPU5	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8166h	TPU5	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8168h	TPU5	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 816Ah	TPU5	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 8170h	TPUB	Timer start register	TSTR	8	8	2, 3 PCLKB	2 ICLK	
0008 8171h	TPUB	Timer synchronous register	TSYR	8	8	2, 3 PCLKB	2 ICLK	
0008 8178h	TPU6	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8179h	TPU7	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 817Ah	TPU8	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 817Bh	TPU9	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 817Ch	TPU10	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 817Dh	TPU11	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8180h	TPU6	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8181h	TPU6	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8182h	TPU6	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	
0008 8183h	TPU6	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	
0008 8184h	TPU6	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8185h	TPU6	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8186h	TPU6	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8188h	TPU6	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 818Ah	TPU6	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 818Ch	TPU6	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	
0008 818Eh	TPU6	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	
0008 8190h	TPU7	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8191h	TPU7	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8192h	TPU7	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 8194h	TPU7	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8195h	TPU7	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8196h	TPU7	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8198h	TPU7	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 819Ah	TPU7	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 81A0h	TPU8	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 81A1h	TPU8	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 81A2h	TPU8	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 81A4h	TPU8	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (16/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 81A5h	TPU8	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 24.
0008 81A6h	TPU8	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 81A8h	TPU8	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 81AAh	TPU8	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 81B0h	TPU9	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 81B1h	TPU9	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 81B2h	TPU9	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	
0008 81B3h	TPU9	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	
0008 81B4h	TPU9	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 81B5h	TPU9	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 81B6h	TPU9	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 81B8h	TPU9	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 81BAh	TPU9	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 81BCh	TPU9	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	
0008 81BEh	TPU9	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	
0008 81C0h	TPU10	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 81C1h	TPU10	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 81C2h	TPU10	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 81C4h	TPU10	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 81C5h	TPU10	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 81C6h	TPU10	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 81C8h	TPU10	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 81CAh	TPU10	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 81D0h	TPU11	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 81D1h	TPU11	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 81D2h	TPU11	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 81D4h	TPU11	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 81D5h	TPU11	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 81D6h	TPU11	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 81D8h	TPU11	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 81DAh	TPU11	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 81E6h	PPG0	PPG output control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 81E7h	PPG0	PPG output mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 81E8h	PPG0	Next data enable register H	NDERH	8	8	2, 3 PCLKB	2 ICLK	
0008 81E9h	PPG0	Next data enable register L	NDERL	8	8	2, 3 PCLKB	2 ICLK	
0008 81EAh	PPG0	Output data register H	PODRH	8	8	2, 3 PCLKB	2 ICLK	
0008 81EBh	PPG0	Output data register L	PODRL	8	8	2, 3 PCLKB	2 ICLK	
0008 81ECh*1	PPG0	Next data register H	NDRH	8	8	2, 3 PCLKB	2 ICLK	
0008 81EDh*2	PPG0	Next data register L	NDRL	8	8	2, 3 PCLKB	2 ICLK	
0008 81EEh*1	PPG0	Next data register H	NDRH2	8	8	2, 3 PCLKB	2 ICLK	
0008 81EFh*2	PPG0	Next data register L	NDRL2	8	8	2, 3 PCLKB	2 ICLK	
0008 81F0h	PPG1	PPG trigger select register	PTRSLR	8	8	2, 3 PCLKB	2 ICLK	
0008 81F6h	PPG1	PPG output control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 81F7h	PPG1	PPG output mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 81F8h	PPG1	Nest data enable register H	NDERH	8	8	2, 3 PCLKB	2 ICLK	
0008 81F9h	PPG1	Nest data enable register L	NDERL	8	8	2, 3 PCLKB	2 ICLK	
0008 81FAh	PPG1	Output data register H	PODRH	8	8	2, 3 PCLKB	2 ICLK	
0008 81FBh	PPG1	Output data register L	PODRL	8	8	2, 3 PCLKB	2 ICLK	
0008 81FCh*3	PPG1	Next data register H	NDRH	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (17/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 81FDh*4	PPG1	Next data register L	NDRL	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 81FEh*3	PPG1	Next data register H	NDRH2	8	8	2, 3 PCLKB	2 ICLK	
0008 81FFh*4	PPG1	Next data register L	NDRL2	8	8	2, 3 PCLKB	2 ICLK	
0008 8200h	TMR0	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 26.
0008 8201h	TMR1	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8202h	TMR0	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8203h	TMR1	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8204h	TMR0	Time constant register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	
0008 8205h	TMR1	Time constant register A	TCORA	8	8*5	2, 3 PCLKB	2 ICLK	
0008 8206h	TMR0	Time constant register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	
0008 8207h	TMR1	Time constant register B	TCORB	8	8*5	2, 3 PCLKB	2 ICLK	
0008 8208h	TMR0	Timer counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 8209h	TMR1	Timer counter	TCNT	8	8*5	2, 3 PCLKB	2 ICLK	
0008 820Ah	TMR0	Timer counter control register	TCCR	8	8	2, 3 PCLKB	2 ICLK	
0008 820Bh	TMR1	Timer counter control register	TCCR	8	8*5	2, 3 PCLKB	2 ICLK	
0008 8210h	TMR2	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8211h	TMR3	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8212h	TMR2	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8213h	TMR3	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8214h	TMR2	Time constant register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	
0008 8215h	TMR3	Time constant register A	TCORA	8	8*5	2, 3 PCLKB	2 ICLK	
0008 8216h	TMR2	Time constant register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	
0008 8217h	TMR3	Time constant register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	
0008 8218h	TMR2	Timer counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 8219h	TMR3	Timer counter	TCNT	8	8*5	2, 3 PCLKB	2 ICLK	
0008 821Ah	TMR2	Timer counter control register	TCCR	8	8	2, 3 PCLKB	2 ICLK	
0008 821Bh	TMR3	Timer counter control register	TCCR	8	8*5	2, 3 PCLKB	2 ICLK	
0008 8280h	CRC	CRC control register	CRCCR	8	8	2, 3 PCLKB	2 ICLK	section 39.
0008 8281h	CRC	CRC data input register	CRCDIR	8	8	2, 3 PCLKB	2 ICLK	
0008 8282h	CRC	CRC data output register	CRCDOR	16	16	2, 3 PCLKB	2 ICLK	
0008 8300h	RIIC0	I^2C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 35.
0008 8301h	RIIC0	I^2C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8302h	RIIC0	I^2C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8303h	RIIC0	I^2C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8304h	RIIC0	I^2C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 8305h	RIIC0	I^2C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK	
0008 8306h	RIIC0	I^2C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK	
0008 8307h	RIIC0	I^2C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8308h	RIIC0	I^2C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8309h	RIIC0	I^2C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	
0008 830Ah	RIIC0	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	
0008 830Bh	RIIC0	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	
0008 830Ch	RIIC0	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	
0008 830Dh	RIIC0	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	
0008 830Eh	RIIC0	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	
0008 830Fh	RIIC0	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	
0008 8310h	RIIC0	I^2C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	
0008 8311h	RIIC0	I^2C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	
0008 8312h	RIIC0	I^2C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	
0008 8313h	RIIC0	I^2C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (18/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 8320h	RIIC1	I²C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 35.
0008 8321h	RIIC1	I²C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8322h	RIIC1	I²C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8323h	RIIC1	I²C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8324h	RIIC1	I²C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 8325h	RIIC1	I²C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK	
0008 8326h	RIIC1	I²C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK	
0008 8327h	RIIC1	I²C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8328h	RIIC1	I²C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8329h	RIIC1	I²C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	
0008 832Ah	RIIC1	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	
0008 832Bh	RIIC1	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	
0008 832Ch	RIIC1	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	
0008 832Dh	RIIC1	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	
0008 832Eh	RIIC1	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	
0008 832Fh	RIIC1	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	
0008 8330h	RIIC1	I²C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	
0008 8331h	RIIC1	I²C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	
0008 8332h	RIIC1	I²C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	
0008 8333h	RIIC1	I²C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	
0008 8340h	RIIC2	I²C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8341h	RIIC2	I²C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8342h	RIIC2	I²C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8343h	RIIC2	I²C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8344h	RIIC2	I²C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 8345h	RIIC2	I²C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK	
0008 8346h	RIIC2	I²C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK	
0008 8347h	RIIC2	I²C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8348h	RIIC2	I²C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8349h	RIIC2	I²C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	
0008 834Ah	RIIC2	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	
0008 834Bh	RIIC2	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	
0008 834Ch	RIIC2	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	
0008 834Dh	RIIC2	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	
0008 834Eh	RIIC2	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	
0008 834Fh	RIIC2	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	
0008 8350h	RIIC2	I²C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	
0008 8351h	RIIC2	I²C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	
0008 8352h	RIIC2	I²C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	
0008 8353h	RIIC2	I²C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	
0008 8360h	RIIC3	I²C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8361h	RIIC3	I²C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8362h	RIIC3	I²C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8363h	RIIC3	I²C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8364h	RIIC3	I²C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 8365h	RIIC3	I²C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK	
0008 8366h	RIIC3	I²C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK	
0008 8367h	RIIC3	I²C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8368h	RIIC3	I²C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	
0008 8369h	RIIC3	I²C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	
0008 836Ah	RIIC3	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	
0008 836Bh	RIIC3	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (19/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 836Ch	RIIC3	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	section 35.
0008 836Dh	RIIC3	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	
0008 836Eh	RIIC3	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	
0008 836Fh	RIIC3	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	
0008 8370h	RIIC3	I^2C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	
0008 8371h	RIIC3	I^2C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	
0008 8372h	RIIC3	I^2C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	
0008 8373h	RIIC3	I^2C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	
0008 8380h	RSPI0	RSPI control register	SPCR	8	8	2, 3 PCLKB	2 ICLK	section 37.
0008 8381h	RSPI0	RSPI slave select polarity register	SSLP	8	8	2, 3 PCLKB	2 ICLK	
0008 8382h	RSPI0	RSPI pin control register	SPPCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8383h	RSPI0	RSPI status register	SPSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8384h	RSPI0	RSPI data register	SPDR	32	16, 32	2, 3 PCLKB	2 ICLK	
0008 8388h	RSPI0	RSPI sequence control register	SPSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8389h	RSPI0	RSPI sequence status register	SPSSR	8	8	2, 3 PCLKB	2 ICLK	
0008 838Ah	RSPI0	RSPI bit rate register	SPBR	8	8	2, 3 PCLKB	2 ICLK	
0008 838Bh	RSPI0	RSPI data control register	SPDCR	8	8	2, 3 PCLKB	2 ICLK	
0008 838Ch	RSPI0	RSPI clock delay register	SPCKD	8	8	2, 3 PCLKB	2 ICLK	
0008 838Dh	RSPI0	RSPI slave select negate delay register	SSLND	8	8	2, 3 PCLKB	2 ICLK	
0008 838Eh	RSPI0	RSPI next access delay register	SPND	8	8	2, 3 PCLKB	2 ICLK	
0008 838Fh	RSPI0	RSPI control register 2	SPCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8390h	RSPI0	RSPI command register 0	SPCMD0	16	16	2, 3 PCLKB	2 ICLK	
0008 8392h	RSPI0	RSPI command register 1	SPCMD1	16	16	2, 3 PCLKB	2 ICLK	
0008 8394h	RSPI0	RSPI command register 2	SPCMD2	16	16	2, 3 PCLKB	2 ICLK	
0008 8396h	RSPI0	RSPI command register 3	SPCMD3	16	16	2, 3 PCLKB	2 ICLK	
0008 8398h	RSPI0	RSPI command register 4	SPCMD4	16	16	2, 3 PCLKB	2 ICLK	
0008 839Ah	RSPI0	RSPI command register 5	SPCMD5	16	16	2, 3 PCLKB	2 ICLK	
0008 839Ch	RSPI0	RSPI command register 6	SPCMD6	16	16	2, 3 PCLKB	2 ICLK	
0008 839Eh	RSPI0	RSPI command register 7	SPCMD7	16	16	2, 3 PCLKB	2 ICLK	
0008 83A0h	RSPI1	RSPI control register	SPCR	8	8	2, 3 PCLKB	2 ICLK	
0008 83A1h	RSPI1	RSPI slave select polarity register	SSLP	8	8	2, 3 PCLKB	2 ICLK	
0008 83A2h	RSPI1	RSPI pin control register	SPPCR	8	8	2, 3 PCLKB	2 ICLK	
0008 83A3h	RSPI1	RSPI status register	SPSR	8	8	2, 3 PCLKB	2 ICLK	
0008 83A4h	RSPI1	RSPI data register	SPDR	32	16, 32	2, 3 PCLKB	2 ICLK	
0008 83A8h	RSPI1	RSPI sequence control register	SPSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 83A9h	RSPI1	RSPI sequence status register	SPSSR	8	8	2, 3 PCLKB	2 ICLK	
0008 83AAh	RSPI1	RSPI bit rate register	SPBR	8	8	2, 3 PCLKB	2 ICLK	
0008 83ABh	RSPI1	RSPI data control register	SPDCR	8	8	2, 3 PCLKB	2 ICLK	
0008 83ACh	RSPI1	RSPI clock delay register	SPCKD	8	8	2, 3 PCLKB	2 ICLK	
0008 83ADh	RSPI1	RSPI slave select negate delay register	SSLND	8	8	2, 3 PCLKB	2 ICLK	
0008 83AEh	RSPI1	RSPI next access delay register	SPND	8	8	2, 3 PCLKB	2 ICLK	
0008 83AFh	RSPI1	RSPI control register 2	SPCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 83B0h	RSPI1	RSPI command register 0	SPCMD0	16	16	2, 3 PCLKB	2 ICLK	
0008 83B2h	RSPI1	RSPI command register 1	SPCMD1	16	16	2, 3 PCLKB	2 ICLK	
0008 83B4h	RSPI1	RSPI command register 2	SPCMD2	16	16	2, 3 PCLKB	2 ICLK	
0008 83B6h	RSPI1	RSPI command register 3	SPCMD3	16	16	2, 3 PCLKB	2 ICLK	
0008 83B8h	RSPI1	RSPI command register 4	SPCMD4	16	16	2, 3 PCLKB	2 ICLK	
0008 83BAh	RSPI1	RSPI command register 5	SPCMD5	16	16	2, 3 PCLKB	2 ICLK	
0008 83BCh	RSPI1	RSPI command register 6	SPCMD6	16	16	2, 3 PCLKB	2 ICLK	
0008 83BEh	RSPI1	RSPI command register 7	SPCMD7	16	16	2, 3 PCLKB	2 ICLK	
0008 83C0h	RSPI2	RSPI control register	SPCR	8	8	2, 3 PCLKB	2 ICLK	
0008 83C1h	RSPI2	RSPI slave select polarity register	SSLP	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (20/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 83C2h	RSPI2	RSPI pin control register	SPPCR	8	8	2, 3 PCLKB	2 ICLK	section 37.
0008 83C3h	RSPI2	RSPI status register	SPSR	8	8	2, 3 PCLKB	2 ICLK	
0008 83C4h	RSPI2	RSPI data register	SPDR	32	16, 32	2, 3 PCLKB	2 ICLK	
0008 83C8h	RSPI2	RSPI sequence control register	SPSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 83C9h	RSPI2	RSPI sequence status register	SPSSR	8	8	2, 3 PCLKB	2 ICLK	
0008 83CAh	RSPI2	RSPI bit rate register	SPBR	8	8	2, 3 PCLKB	2 ICLK	
0008 83CBh	RSPI2	RSPI data control register	SPDCR	8	8	2, 3 PCLKB	2 ICLK	
0008 83CCh	RSPI2	RSPI clock delay register	SPCKD	8	8	2, 3 PCLKB	2 ICLK	
0008 83CDh	RSPI2	RSPI slave select negate delay register	SSLND	8	8	2, 3 PCLKB	2 ICLK	
0008 83CEh	RSPI2	RSPI next access delay register	SPND	8	8	2, 3 PCLKB	2 ICLK	
0008 83CFh	RSPI2	RSPI control register 2	SPCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 83D0h	RSPI2	RSPI command register 0	SPCMD0	16	16	2, 3 PCLKB	2 ICLK	
0008 83D2h	RSPI2	RSPI command register 1	SPCMD1	16	16	2, 3 PCLKB	2 ICLK	
0008 83D4h	RSPI2	RSPI command register 2	SPCMD2	16	16	2, 3 PCLKB	2 ICLK	
0008 83D6h	RSPI2	RSPI command register 3	SPCMD3	16	16	2, 3 PCLKB	2 ICLK	
0008 83D8h	RSPI2	RSPI command register 4	SPCMD4	16	16	2, 3 PCLKB	2 ICLK	
0008 83DAh	RSPI2	RSPI command register 5	SPCMD5	16	16	2, 3 PCLKB	2 ICLK	
0008 83DCh	RSPI2	RSPI command register 6	SPCMD6	16	16	2, 3 PCLKB	2 ICLK	
0008 83DEh	RSPI2	RSPI command register 7	SPCMD7	16	16	2, 3 PCLKB	2 ICLK	
0008 8600h	MTU3	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 22.
0008 8601h	MTU4	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8602h	MTU3	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8603h	MTU4	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8604h	MTU3	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	
0008 8605h	MTU3	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	
0008 8606h	MTU4	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	
0008 8607h	MTU4	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	
0008 8608h	MTU3	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8609h	MTU4	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 860Ah	MTU	Timer output master enable register	TOER	8	8	2, 3 PCLKB	2 ICLK	
0008 860Dh	MTU	Timer gate control register	TGCR	8	8	2, 3 PCLKB	2 ICLK	
0008 860Eh	MTU	Timer output control register 1	TOCR1	8	8	2, 3 PCLKB	2 ICLK	
0008 860Fh	MTU	Timer output control register 2	TOCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 8610h	MTU3	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8612h	MTU4	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8614h	MTU	Timer cycle data register	TCDR	16	16	2, 3 PCLKB	2 ICLK	
0008 8616h	MTU	Timer dead time data register	TDDR	16	16	2, 3 PCLKB	2 ICLK	
0008 8618h	MTU3	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 861Ah	MTU3	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 861Ch	MTU4	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 861Eh	MTU4	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 8620h	MTU	Timer subcounter	TCNTS	16	16	2, 3 PCLKB	2 ICLK	
0008 8622h	MTU	Timer cycle buffer register	TCBR	16	16	2, 3 PCLKB	2 ICLK	
0008 8624h	MTU3	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	
0008 8626h	MTU3	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	
0008 8628h	MTU4	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	
0008 862Ah	MTU4	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	
0008 862Ch	MTU3	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 862Dh	MTU4	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8630h	MTU	Timer interrupt skipping set register	TITCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8631h	MTU	Timer interrupt skipping counter	TITCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 8632h	MTU	Timer buffer transfer set register	TBTER	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (21/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 8634h	MTU	Timer dead time enable register	TDER	8	8	2, 3 PCLKB	2 ICLK	section 22.
0008 8636h	MTU	Timer output level buffer register	TOLBR	8	8	2, 3 PCLKB	2 ICLK	
0008 8638h	MTU3	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK	
0008 8639h	MTU4	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK	
0008 8640h	MTU4	Timer A/D converter start request control register	TADCR	16	16	2, 3 PCLKB	2 ICLK	
0008 8644h	MTU4	Timer A/D converter start request cycle set register A	TADCORA	16	16	2, 3 PCLKB	2 ICLK	
0008 8646h	MTU4	Timer A/D converter start request cycle set register B	TADCORB	16	16	2, 3 PCLKB	2 ICLK	
0008 8648h	MTU4	Timer A/D converter start request cycle set buffer register A	TADCOBRA	16	16	2, 3 PCLKB	2 ICLK	
0008 864Ah	MTU4	Timer A/D converter start request cycle set buffer register B	TADCOBRB	16	16	2, 3 PCLKB	2 ICLK	
0008 8660h	MTU	Timer waveform control register	TWCR	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8680h	MTU	Timer start register	TSTR	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8681h	MTU	Timer synchronous register	TSYR	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8684h	MTU	Timer read/write enable register	TRWER	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8690h	MTU0	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8691h	MTU1	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8692h	MTU2	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8693h	MTU3	Noise filter control register	NFCR3	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8694h	MTU4	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8695h	MTU5	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 8700h	MTU0	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8701h	MTU0	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8702h	MTU0	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	
0008 8703h	MTU0	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	
0008 8704h	MTU0	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8705h	MTU0	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8706h	MTU0	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8708h	MTU0	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 870Ah	MTU0	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 870Ch	MTU0	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	
0008 870Eh	MTU0	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	
0008 8720h	MTU0	Timer general register E	TGRE	16	16	2, 3 PCLKB	2 ICLK	
0008 8722h	MTU0	Timer general register F	TGRF	16	16	2, 3 PCLKB	2 ICLK	
0008 8724h	MTU0	Timer interrupt enable register2	TIER2	8	8	2, 3 PCLKB	2 ICLK	
0008 8726h	MTU0	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK	
0008 8780h	MTU1	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8781h	MTU1	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8782h	MTU1	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 8784h	MTU1	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8785h	MTU1	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8786h	MTU1	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8788h	MTU1	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 878Ah	MTU1	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	
0008 8790h	MTU1	Timer input capture control register	TICCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8800h	MTU2	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 8801h	MTU2	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	
0008 8802h	MTU2	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	
0008 8804h	MTU2	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 8805h	MTU2	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	
0008 8806h	MTU2	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 8808h	MTU2	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	
0008 880Ah	MTU2	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (22/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 8880h	MTU5	Timer counter U	TCNTU	16	16	2, 3 PCLKB	2 ICLK	section 22.
0008 8882h	MTU5	Timer general register U	TGRU	16	16	2, 3 PCLKB	2 ICLK	
0008 8884h	MTU5	Timer control register U	TCRU	8	8	2, 3 PCLKB	2 ICLK	
0008 8886h	MTU5	Timer I/O control register U	TIORU	8	8	2, 3 PCLKB	2 ICLK	
0008 8890h	MTU5	Timer counter V	TCNTV	16	16	2, 3 PCLKB	2 ICLK	
0008 8892h	MTU5	Timer general register V	TGRV	16	16	2, 3 PCLKB	2 ICLK	
0008 8894h	MTU5	Timer control register V	TCRV	8	8	2, 3 PCLKB	2 ICLK	
0008 8896h	MTU5	Timer I/O control register V	TIORV	8	8	2, 3 PCLKB	2 ICLK	
0008 88A0h	MTU5	Timer counter W	TCNTW	16	16	2, 3 PCLKB	2 ICLK	
0008 88A2h	MTU5	Timer general register W	TGRW	16	16	2, 3 PCLKB	2 ICLK	
0008 88A4h	MTU5	Timer control register W	TCRW	8	8	2, 3 PCLKB	2 ICLK	
0008 88A6h	MTU5	Timer I/O control register W	TIORW	8	8	2, 3 PCLKB	2 ICLK	
0008 88B2h	MTU5	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	
0008 88B4h	MTU5	Timer start register	TSTR	8	8	2, 3 PCLKB	2 ICLK	
0008 88B6h	MTU5	Timer compare match clear register	TCNTCMPCLR	8	8	2, 3 PCLKB	2 ICLK	
0008 8900h	POE	Input level control/status register 1	ICSR1	16	16	2, 3 PCLKB	2 ICLK	section 23.
0008 8902h	POE	Output level control/status register 1	OCSR1	16	16	2, 3 PCLKB	2 ICLK	
0008 8908h	POE	Input level control/status register 2	ICSR2	16	16	2, 3 PCLKB	2 ICLK	
0008 890Ah	POE	Software port output enable register	SPOER	8	8	2, 3 PCLKB	2 ICLK	
0008 890Bh	POE	Port output enable control register 1	POECR1	8	8	2, 3 PCLKB	2 ICLK	
0008 890Ch	POE	Port output enable control register 2	POECR2	8	8	2, 3 PCLKB	2 ICLK	
0008 890Eh	POE	Input level control/status register 3	ICSR3	16	16	2, 3 PCLKB	2 ICLK	
0008 9000h	S12AD	A/D control register	ADCSR	8	8	2, 3 PCLKB	2 ICLK	section 40.
0008 9004h	S12AD	A/D channel select register 0	ADANS0	16	16	2, 3 PCLKB	2 ICLK	
0008 9006h	S12AD	A/D channel select register 1	ADANS1	16	16	2, 3 PCLKB	2 ICLK	
0008 9008h	S12AD	A/D-converted value addition mode select register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	
0008 900Ah	S12AD	A/D-converted value addition mode select register 1	ADADS1	16	16	2, 3 PCLKB	2 ICLK	
0008 900Ch	S12AD	A/D-converted value addition count select register	ADADC	8	8	2, 3 PCLKB	2 ICLK	
0008 900Eh	S12AD	A/D control extended register	ADCER	16	16	2, 3 PCLKB	2 ICLK	
0008 9010h	S12AD	A/D start trigger select register	ADSTRGR	8	8	2, 3 PCLKB	2 ICLK	
0008 9012h	S12AD	A/D-converted extended input control register	ADEXICR	16	16	2, 3 PCLKB	2 ICLK	
0008 901Ah	S12AD	A/D temperature sensor data register	ADTSDR	16	16	2, 3 PCLKB	2 ICLK	
0008 901Ch	S12AD	A/D internal reference voltage data register	ADOCDR	16	16	2, 3 PCLKB	2 ICLK	
0008 9020h	S12AD	A/D data register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	
0008 9022h	S12AD	A/D data register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	
0008 9024h	S12AD	A/D data register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	
0008 9026h	S12AD	A/D data register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	
0008 9028h	S12AD	A/D data register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	
0008 902Ah	S12AD	A/D data register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	
0008 902Ch	S12AD	A/D data register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	
0008 902Eh	S12AD	A/D data register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	
0008 9030h	S12AD	A/D data register 8	ADDR8	16	16	2, 3 PCLKB	2 ICLK	
0008 9032h	S12AD	A/D data register 9	ADDR9	16	16	2, 3 PCLKB	2 ICLK	
0008 9034h	S12AD	A/D data register 10	ADDR10	16	16	2, 3 PCLKB	2 ICLK	
0008 9036h	S12AD	A/D data register 11	ADDR11	16	16	2, 3 PCLKB	2 ICLK	
0008 9038h	S12AD	A/D data register 12	ADDR12	16	16	2, 3 PCLKB	2 ICLK	
0008 903Ah	S12AD	A/D data register 13	ADDR13	16	16	2, 3 PCLKB	2 ICLK	
0008 903Ch	S12AD	A/D data register 14	ADDR14	16	16	2, 3 PCLKB	2 ICLK	
0008 903Eh	S12AD	A/D data register 15	ADDR15	16	16	2, 3 PCLKB	2 ICLK	
0008 9040h	S12AD	A/D data register 16	ADDR16	16	16	2, 3 PCLKB	2 ICLK	
0008 9042h	S12AD	A/D data register 17	ADDR17	16	16	2, 3 PCLKB	2 ICLK	
0008 9044h	S12AD	A/D data register 18	ADDR18	16	16	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (23/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 9046h	S12AD	A/D data register 19	ADDR19	16	16	2, 3 PCLKB	2 ICLK	section 40.
0008 9048h	S12AD	A/D data register 20	ADDR20	16	16	2, 3 PCLKB	2 ICLK	
0008 9060h	S12AD	A/D sampling state register01	ADSSTR01	16	16	2, 3 PCLKB	2 ICLK	
0008 9070h	S12AD	A/D sampling state register 23	ADSSTR23	16	16	2, 3 PCLKB	2 ICLK	
0008 9800h	AD	A/D data register A	ADDRA	16	16	2, 3 PCLKB	2 ICLK	section 41.
0008 9802h	AD	A/D data register B	ADDRB	16	16	2, 3 PCLKB	2 ICLK	
0008 9804h	AD	A/D data register C	ADDRC	16	16	2, 3 PCLKB	2 ICLK	
0008 9806h	AD	A/D data register D	ADDRD	16	16	2, 3 PCLKB	2 ICLK	
0008 9808h	AD	A/D data register E	ADDRE	16	16	2, 3 PCLKB	2 ICLK	
0008 980Ah	AD	A/D data register F	ADDRF	16	16	2, 3 PCLKB	2 ICLK	
0008 980Ch	AD	A/D data register G	ADDRG	16	16	2, 3 PCLKB	2 ICLK	
0008 980Eh	AD	A/D data register H	ADDRH	16	16	2, 3 PCLKB	2 ICLK	
0008 9810h	AD	A/D control/status register	ADCSR	8	8	2, 3 PCLKB	2 ICLK	
0008 9811h	AD	A/D control register	ADCR	8	8	2, 3 PCLKB	2 ICLK	
0008 9812h	AD	A/D control register 2	ADCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 9813h	AD	A/D sampling state register	ADSSTR	8	8	2, 3 PCLKB	2 ICLK	
0008 981Fh	AD	A/D self-diagnostic register	ADDIAGR	8	8	2, 3 PCLKB	2 ICLK	
0008 A000h	SCI0	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 34.
0008 A001h	SCI0	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A002h	SCI0	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A003h	SCI0	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A004h	SCI0	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A005h	SCI0	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A006h	SCI0	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A007h	SCI0	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A008h	SCI0	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A009h	SCI0	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A00Ah	SCI0	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A00Bh	SCI0	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A00Ch	SCI0	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A00Dh	SCI0	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A020h	SCI1	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A021h	SCI1	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A022h	SCI1	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A023h	SCI1	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A024h	SCI1	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A025h	SCI1	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A026h	SCI1	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A027h	SCI1	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A028h	SCI1	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A029h	SCI1	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A02Ah	SCI1	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A02Bh	SCI1	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A02Ch	SCI1	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A02Dh	SCI1	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A040h	SCI2	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A041h	SCI2	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A042h	SCI2	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A043h	SCI2	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A044h	SCI2	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A045h	SCI2	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A046h	SCI2	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (24/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 A047h	SCI2	Serial extended mode register	SEM	8	8	2, 3 PCLKB	2 ICLK	section 34.
0008 A048h	SCI2	Noise filter setting register	SNFR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A049h	SCI2	I2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A04Ah	SCI2	I2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A04Bh	SCI2	I2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A04Ch	SCI2	I2C status register	SIS	8	8	2, 3 PCLKB	2 ICLK	
0008 A04Dh	SCI2	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A060h	SCI3	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A061h	SCI3	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A062h	SCI3	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A063h	SCI3	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A064h	SCI3	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A065h	SCI3	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A066h	SCI3	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A067h	SCI3	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A068h	SCI3	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A069h	SCI3	I2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A06Ah	SCI3	I2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A06Bh	SCI3	I2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A06Ch	SCI3	I2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A06Dh	SCI3	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A080h	SCI4	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A081h	SCI4	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A082h	SCI4	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A083h	SCI4	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A084h	SCI4	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A085h	SCI4	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A086h	SCI4	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A087h	SCI4	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A088h	SCI4	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A089h	SCI4	I2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A08Ah	SCI4	I2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A08Bh	SCI4	I2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A08Ch	SCI4	I2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A08Dh	SCI4	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A0h	SCI5	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A1h	SCI5	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A2h	SCI5	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A3h	SCI5	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A4h	SCI5	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A5h	SCI5	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A6h	SCI5	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A7h	SCI5	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A8h	SCI5	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0A9h	SCI5	I2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A0AAh	SCI5	I2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A0ABh	SCI5	I2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A0ACh	SCI5	I2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0ADh	SCI5	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0C0h	SCI6	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0C1h	SCI6	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0C2h	SCI6	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (25/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 A0C3h	SCI6	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 34.
0008 A0C4h	SCI6	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0C5h	SCI6	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0C6h	SCI6	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0C7h	SCI6	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0C8h	SCI6	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0C9h	SCI6	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A0CAh	SCI6	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A0CBh	SCI6	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A0CCh	SCI6	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0CDh	SCI6	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E0h	SCI7	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E1h	SCI7	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E2h	SCI7	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E3h	SCI7	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E4h	SCI7	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E5h	SCI7	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E6h	SCI7	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E7h	SCI7	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E8h	SCI7	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0E9h	SCI7	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A0EAh	SCI7	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A0EBh	SCI7	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A0ECh	SCI7	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A0EDh	SCI7	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A100h	SCI8	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A101h	SCI8	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A102h	SCI8	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A103h	SCI8	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A104h	SCI8	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A105h	SCI8	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A106h	SCI8	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A107h	SCI8	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A108h	SCI8	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A109h	SCI8	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A10Ah	SCI8	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A10Bh	SCI8	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A10Ch	SCI8	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A10Dh	SCI8	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A120h	SCI9	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A121h	SCI9	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A122h	SCI9	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A123h	SCI9	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A124h	SCI9	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A125h	SCI9	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A126h	SCI9	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A127h	SCI9	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A128h	SCI9	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A129h	SCI9	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A12Ah	SCI9	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A12Bh	SCI9	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A12Ch	SCI9	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (26/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 A12Dh	SCI9	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 34.
0008 A140h	SCI10	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A141h	SCI10	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A142h	SCI10	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A143h	SCI10	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A144h	SCI10	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A145h	SCI10	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A146h	SCI10	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A147h	SCI10	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A148h	SCI10	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A149h	SCI10	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A14Ah	SCI10	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A14Bh	SCI10	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A14Ch	SCI10	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A14Dh	SCI10	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A160h	SCI11	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A161h	SCI11	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	
0008 A162h	SCI11	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	
0008 A163h	SCI11	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A164h	SCI11	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	
0008 A165h	SCI11	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	
0008 A166h	SCI11	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A167h	SCI11	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A168h	SCI11	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	
0008 A169h	SCI11	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	
0008 A16Ah	SCI11	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	
0008 A16Bh	SCI11	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	
0008 A16Ch	SCI11	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	
0008 A16Dh	SCI11	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 A800h	IEB	IEBus control register	IECTR	8	8	3 to 4 PCLKB	2, 3 ICLK	section 38.
0008 A801h	IEB	IEBus command register	IECMR	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A802h	IEB	IEBus master control register	IEMCR	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A803h	IEB	IEBus master unit address register 1	IEAR1	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A804h	IEB	IEBus master unit address register 2	IEAR2	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A805h	IEB	IEBus slave address setting register 1	IESA1	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A806h	IEB	IEBus slave address setting register 2	IESA2	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A807h	IEB	IEBus transmit message length register	IETBFL	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A809h	IEB	IEBus reception master address register 1	IEMA1	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A80Ah	IEB	IEBus reception master address register 2	IEMA2	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A80Bh	IEB	IEBus receive control field register	IERCTL	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A80Ch	IEB	IEBus receive message length register	IERBFL	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A80Eh	IEB	IEBus lock address register 1	IELA1	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A80Fh	IEB	IEBus lock address register 2	IELA2	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A810h	IEB	IEBus general flag register	IEFLG	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A811h	IEB	IEBus transmit status register	IETSR	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A812h	IEB	IEBus transmit interrupt enable register	IEIET	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A814h	IEB	IEBus receive status register	IERSR	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A815h	IEB	IEBus receive interrupt enable register	IEIER	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A818h	IEB	IEBus clock select register	IECKSR	8	8	3 to 4 PCLKB	2, 3 ICLK	
0008 A900h to 0008 A91Fh	IEB	IEBus transmit data buffer register 001 to 032	IETB001 to 032	8	8	3 to 4 PCLKB	2, 3 ICLK	

Table 5.1 List of I/O Registers (Address Order) (27/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 AA00h to 0008 AA1Fh	IEB	IEBus receive data buffer register 001 to 032	IERB001 to 032	8	8	3 to 4 PCLKB	2, 3 ICLK	section 38.
0008 B300h	SCI12	Serial mode register	SMR12	8	8	2, 3 PCLKB	2 ICLK	section 34.
0008 B301h	SCI12	Bit rate register	BRR12	8	8	2, 3 PCLKB	2 ICLK	
0008 B302h	SCI12	Serial control register	SCR12	8	8	2, 3 PCLKB	2 ICLK	
0008 B303h	SCI12	Transmit data register	TDR12	8	8	2, 3 PCLKB	2 ICLK	
0008 B304h	SCI12	Serial status register	SSR12	8	8	2, 3 PCLKB	2 ICLK	
0008 B305h	SCI12	Receive data register	RDR12	8	8	2, 3 PCLKB	2 ICLK	
0008 B306h	SCI12	Smart card mode register	SCMR12	8	8	2, 3 PCLKB	2 ICLK	
0008 B307h	SCI12	Serial extended mode register	SEMR12	8	8	2, 3 PCLKB	2 ICLK	
0008 B308h	SCI12	Noise filter setting register	SNFR12	8	8	2, 3 PCLKB	2 ICLK	
0008 B309h	SCI12	I^2C mode register 1	SIMR112	8	8	2, 3 PCLKB	2 ICLK	
0008 B30Ah	SCI12	I^2C mode register 2	SIMR212	8	8	2, 3 PCLKB	2 ICLK	
0008 B30Bh	SCI12	I^2C mode register 3	SIMR312	8	8	2, 3 PCLKB	2 ICLK	
0008 B30Ch	SCI12	I^2C status register	SIS12	8	8	2, 3 PCLKB	2 ICLK	
0008 B30Dh	SCI12	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	
0008 B320h	SCI12	Extended serial module enable register	ESMER	8	8	2, 3 PCLKB	2 ICLK	
0008 B321h	SCI12	Control register 0	CR0	8	8	2, 3 PCLKB	2 ICLK	
0008 B322h	SCI12	Control register 1	CR1	8	8	2, 3 PCLKB	2 ICLK	
0008 B323h	SCI12	Control register 2	CR2	8	8	2, 3 PCLKB	2 ICLK	
0008 B324h	SCI12	Control register 3	CR3	8	8	2, 3 PCLKB	2 ICLK	
0008 B325h	SCI12	Port control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 B326h	SCI12	Interrupt control register	ICR	8	8	2, 3 PCLKB	2 ICLK	
0008 B327h	SCI12	Status register	STR	8	8	2, 3 PCLKB	2 ICLK	
0008 B328h	SCI12	Status clear register	STCR	8	8	2, 3 PCLKB	2 ICLK	
0008 B329h	SCI12	Control field 0 data register	CF0DR	8	8	2, 3 PCLKB	2 ICLK	
0008 B32Ah	SCI12	Control field 0 compare enable register	CF0CR	8	8	2, 3 PCLKB	2 ICLK	
0008 B32Bh	SCI12	Control field 0 receive data register	CF0RR	8	8	2, 3 PCLKB	2 ICLK	
0008 B32Ch	SCI12	Primary control field 1 data register	PCF1DR	8	8	2, 3 PCLKB	2 ICLK	
0008 B32Dh	SCI12	Secondary control field 1 data register	SCF1DR	8	8	2, 3 PCLKB	2 ICLK	
0008 B32Eh	SCI12	Control field 1 compare enable register	CF1CR	8	8	2, 3 PCLKB	2 ICLK	
0008 B32Fh	SCI12	Control field 1 receive data register	CF1RR	8	8	2, 3 PCLKB	2 ICLK	
0008 B330h	SCI12	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0008 B331h	SCI12	Timer mode register	TMR	8	8	2, 3 PCLKB	2 ICLK	
0008 B332h	SCI12	Timer prescaler register	TPRE	8	8	2, 3 PCLKB	2 ICLK	
0008 B333h	SCI12	Timer count register	TCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 C000h	PORT0	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C001h	PORT1	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C002h	PORT2	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C003h	PORT3	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C004h	PORT4	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C005h	PORT5	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C006h	PORT6	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C007h	PORT7	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C008h	PORT8	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C009h	PORT9	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C00Ah	PORTA	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C00Bh	PORTB	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C00Ch	PORTC	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C00Dh	PORTD	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C00Eh	PORTE	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C00Fh	PORTF	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (28/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 C010h	PORTG	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C012h	PORTJ	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C020h	PORT0	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C021h	PORT1	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C022h	PORT2	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C023h	PORT3	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C024h	PORT4	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C025h	PORT5	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C026h	PORT6	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C027h	PORT7	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C028h	PORT8	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C029h	PORT9	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C02Ah	PORTA	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C02Bh	PORTB	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C02Ch	PORTC	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C02Dh	PORTD	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C02Eh	PORTE	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C02Fh	PORTF	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C030h	PORTG	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C032h	PORTJ	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	
0008 C040h	PORT0	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C041h	PORT1	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C042h	PORT2	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C043h	PORT3	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C044h	PORT4	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C045h	PORT5	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C046h	PORT6	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C047h	PORT7	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C048h	PORT8	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C049h	PORT9	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C04Ah	PORTA	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C04Bh	PORTB	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C04Ch	PORTC	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C04Dh	PORTD	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C04Eh	PORTE	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C04Fh	PORTF	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C050h	PORTG	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C052h	PORTJ	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	
0008 C060h	PORT0	Port input data register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C061h	PORT1	Port input data register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C062h	PORT2	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C063h	PORT3	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C064h	PORT4	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C065h	PORT5	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C066h	PORT6	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C067h	PORT7	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C068h	PORT8	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C069h	PORT9	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C06Ah	PORTA	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C06Bh	PORTB	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C06Ch	PORTC	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C06Dh	PORTD	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (29/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 C06Eh	PORTE	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C06Fh	PORTF	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C070h	PORTG	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C072h	PORTJ	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	
0008 C080h	PORT0	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C081h	PORT0	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C082h	PORT1	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C083h	PORT1	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C084h	PORT2	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C085h	PORT2	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C086h	PORT3	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C087h	PORT3	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C088h	PORT4	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C089h	PORT4	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C08Ah	PORT5	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C08Bh	PORT5	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C08Ch	PORT6	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C08Dh	PORT6	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C08Eh	PORT7	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C08Fh	PORT7	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C090h	PORT8	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C091h	PORT8	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C092h	PORT9	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C093h	PORT9	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C094h	PORTA	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C095h	PORTA	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C096h	PORTB	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C097h	PORTB	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C098h	PORTC	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C099h	PORTC	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C09Ah	PORTD	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C09Bh	PORTD	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C09Ch	PORTE	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C09Dh	PORTE	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C09Eh	PORTF	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C09Fh	PORTF	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C0A0h	PORTG	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C0A1h	PORTG	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C0A4h	PORTJ	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C0A5h	PORTJ	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C0C0h	PORT0	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C1h	PORT1	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C2h	PORT2	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C3h	PORT3	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C4h	PORT4	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C5h	PORT5	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C6h	PORT6	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C7h	PORT7	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C8h	PORT8	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0C9h	PORT9	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0CAh	PORTA	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0CBh	PORTB	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (30/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 C0CCh	PORTC	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C0CDh	PORTD	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0CEh	PORTE	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0CFh	PORTF	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0D0h	PORTG	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0D2h	PORTJ	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0E0h	PORT0	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0E2h	PORT2	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0E5h	PORT5	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0E9h	PORT9	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0EAh	PORTA	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0EBh	PORTB	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0ECh	PORTC	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0EDh	PORTD	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0EEh	PORTE	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C0F0h	PORTG	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	
0008 C100h	MPC	CS output enable register	PFCSE	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 C102h	MPC	CS output pin select register 0	PFCSS0	8	8	2, 3 PCLKB	2 ICLK	
0008 C103h	MPC	CS output pin select register 1	PFCSS1	8	8	2, 3 PCLKB	2 ICLK	
0008 C104h	MPC	Address output enable register 0	PFAOE0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C105h	MPC	Address output enable register 1	PFAOE1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C106h	MPC	External bus control register 0	PFBCR0	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C107h	MPC	External bus control register 1	PFBCR1	8	8, 16	2, 3 PCLKB	2 ICLK	
0008 C10Eh	MPC	Ethernet control resister 1	PFENET	8	8	2, 3 PCLKB	2 ICLK	
0008 C114h	MPC	USB0 control register	PFUSB0	8	8	2, 3 PCLKB	2 ICLK	
0008 C115h	MPC	USB1 control register	PFUSB1	8	8	2, 3 PCLKB	2 ICLK	
0008 C11Fh	MPC	Write protection register	PWPR	8	8	2, 3 PCLKB	2 ICLK	
0008 C140h	MPC	P00 pin function control register	P00PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C141h	MPC	P01 pin function control register	P01PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C142h	MPC	P02 pin function control register	P02PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C143h	MPC	P03 pin function control register	P03PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C145h	MPC	P05 pin function control register	P05PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C147h	MPC	P07 pin function control register	P07PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C148h	MPC	P10 pin function control register	P10PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C149h	MPC	P11 pin function control register	P11PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C14Ah	MPC	P12 pin function control register	P12PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C14Bh	MPC	P13 pin function control register	P13PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C14Ch	MPC	P14 pin function control register	P14PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C14Dh	MPC	P15 pin function control register	P15PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C14Eh	MPC	P16 pin function control register	P16PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C14Fh	MPC	P17 pin function control register	P17PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C150h	MPC	P20 pin function control register	P20PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C151h	MPC	P21 pin function control register	P21PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C152h	MPC	P22 pin function control register	P22PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C153h	MPC	P23 pin function control register	P23PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C154h	MPC	P24 pin function control register	P24PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C155h	MPC	P25 pin function control register	P25PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C156h	MPC	P26 pin function control register	P26PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C157h	MPC	P27 pin function control register	P27PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C158h	MPC	P30 pin function control register	P30PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C159h	MPC	P31 pin function control register	P31PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C15Ah	MPC	P32 pin function control register	P32PFS	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (31/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 C15Bh	MPC	P33 pin function control register	P33PFS	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 C15Ch	MPC	P34 pin function control register	P34PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C160h	MPC	P40 pin function control register	P40PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C161h	MPC	P41 pin function control register	P41PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C162h	MPC	P42 pin function control register	P42PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C163h	MPC	P43 pin function control register	P43PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C164h	MPC	P44 pin function control register	P44PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C165h	MPC	P45 pin function control register	P45PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C166h	MPC	P46 pin function control register	P46PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C167h	MPC	P47 pin function control register	P47PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C168h	MPC	P50 pin function control register	P50PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C169h	MPC	P51 pin function control register	P51PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C16Ah	MPC	P52 pin function control register	P52PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C16Ch	MPC	P54 pin function control register	P54PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C16Dh	MPC	P55 pin function control register	P55PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C16Eh	MPC	P56 pin function control register	P56PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C16Fh	MPC	P57 pin function control register	P57PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C170h	MPC	P60 pin function control register	P60PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C171h	MPC	P61 pin function control register	P61PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C176h	MPC	P66 pin function control register	P66PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C177h	MPC	P67 pin function control register	P67PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C178h	MPC	P70 pin function control register	P70PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C179h	MPC	P71 pin function control register	P71PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C17Ah	MPC	P72 pin function control register	P72PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C17Bh	MPC	P73 pin function control register	P73PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C17Ch	MPC	P74 pin function control register	P74PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C17Dh	MPC	P75 pin function control register	P75PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C17Eh	MPC	P76 pin function control register	P76PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C17Fh	MPC	P77 pin function control register	P77PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C180h	MPC	P80 pin function control register	P80PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C181h	MPC	P81 pin function control register	P81PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C182h	MPC	P82 pin function control register	P82PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C183h	MPC	P83 pin function control register	P83PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C186h	MPC	P86 pin function control register	P86PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C187h	MPC	P87 pin function control register	P87PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C188h	MPC	P90 pin function control register	P90PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C189h	MPC	P91 pin function control register	P91PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C18Ah	MPC	P92 pin function control register	P92PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C18Bh	MPC	P93 pin function control register	P93PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C190h	MPC	PA0 pin function control register	PA0PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C191h	MPC	PA1 pin function control register	PA1PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C192h	MPC	PA2 pin function control register	PA2PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C193h	MPC	PA3 pin function control register	PA3PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C194h	MPC	PA4 pin function control register	PA4PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C195h	MPC	PA5 pin function control register	PA5PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C196h	MPC	PA6 pin function control register	PA6PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C197h	MPC	PA7 pin function control register	PA7PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C198h	MPC	PB0 pin function control register	PB0PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C199h	MPC	PB1 pin function control register	PB1PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C19Ah	MPC	PB2 pin function control register	PB2PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C19Bh	MPC	PB3 pin function control register	PB3PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C19Ch	MPC	PB4 pin function control register	PB4PFS	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (32/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 C19Dh	MPC	PB5 pin function control register	PB5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 C19Eh	MPC	PB6 pin function control register	PB6PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C19Fh	MPC	PB7 pin function control register	PB7PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A0h	MPC	PC0 pin function control register	PC0PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A1h	MPC	PC1 pin function control register	PC1PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A2h	MPC	PC2 pin function control register	PC2PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A3h	MPC	PC3 pin function control register	PC3PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A4h	MPC	PC4 pin function control register	PC4PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A5h	MPC	PC5 pin function control register	PC5PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A6h	MPC	PC6 pin function control register	PC6PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A7h	MPC	PC7 pin function control register	PC7PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A8h	MPC	PD0 pin function control register	PD0PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1A9h	MPC	PD1 pin function control register	PD1PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1AAh	MPC	PD2 pin function control register	PD2PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1ABh	MPC	PD3 pin function control register	PD3PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1ACh	MPC	PD4 pin function control register	PD4PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1ADh	MPC	PD5 pin function control register	PD5PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1AEh	MPC	PD6 pin function control register	PD6PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1AFh	MPC	PD7 pin function control register	PD7PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B0h	MPC	PE0 pin function control register	PE0PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B1h	MPC	PE1 pin function control register	PE1PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B2h	MPC	PE2 pin function control register	PE2PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B3h	MPC	PE3 pin function control register	PE3PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B4h	MPC	PE4 pin function control register	PE4PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B5h	MPC	PE5 pin function control register	PE5PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B6h	MPC	PE6 pin function control register	PE6PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B7h	MPC	PE7 pin function control register	PE7PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B8h	MPC	PF0 pin function control register	PF0PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1B9h	MPC	PF1 pin function control register	PF1PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1BAh	MPC	PF2 pin function control register	PF2PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1BDh	MPC	PF5 pin function control register	PF5PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C1D3h	MPC	PJ3 pin function control register	PJ3PFS	8	8	2, 3 PCLKB	2 ICLK	
0008 C280h	SYSTEM	Deep standby control register	DPSBYCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 11.
0008 C282h	SYSTEM	Deep standby interrupt enable register 0	DPSIER0	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C283h	SYSTEM	Deep standby interrupt enable register 1	DPSIER1	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C284h	SYSTEM	Deep standby interrupt enable register 2	DPSIER2	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C285h	SYSTEM	Deep standby interrupt enable register 3	DPSIER3	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C286h	SYSTEM	Deep standby interrupt flag register 0	DPSIFR0	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C287h	SYSTEM	Deep standby interrupt flag register 1	DPSIFR1	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C288h	SYSTEM	Deep standby interrupt flag register 2	DPSIFR2	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C289h	SYSTEM	Deep standby interrupt flag register 3	DPSIFR3	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C28Ah	SYSTEM	Deep standby interrupt edge register 0	DPSIEGR0	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C28Bh	SYSTEM	Deep standby interrupt edge register 1	DPSIEGR1	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C28Ch	SYSTEM	Deep standby interrupt edge register 2	DPSIEGR2	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C28Dh	SYSTEM	Deep standby interrupt edge register 3	DPSIEGR3	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C290h	SYSTEM	Reset status register 0	RSTSR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 6.
0008 C291h	SYSTEM	Reset status register 1	RSTSR1	8	8	4, 5 PCLKB	2, 3 ICLK	section 6.
0008 C293h	SYSTEM	Main clock oscillator forced oscillation control register	MOFCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 9.
0008 C294h	SYSTEM	High-speed on-chip oscillator power control register	HOCOPCR	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C296h	FLASH	Flash write/erase protect register	FWEPROR	8	8	4, 5 PCLKB	2, 3 ICLK	section 45.
0008 C297h	SYSTEM	Voltage monitoring circuit control register	LVCMPCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 8.
0008 C298h	SYSTEM	Voltage detection level select register	LVDLVLR	8	8	4, 5 PCLKB	2, 3 ICLK	

Table 5.1 List of I/O Registers (Address Order) (33/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 C29Ah	SYSTEM	Voltage monitoring 1 circuit control register 0	LVD1CR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 8.
0008 C29Bh	SYSTEM	Voltage monitoring 2 circuit control register 0	LVD2CR0	8	8	4, 5 PCLKB	2, 3 ICLK	
0008 C2A0h to 0008 C2BFh	SYSTEM	Deep standby backup register 0 to 31	DPSBKR0 to 31	8	8	4, 5 PCLKB	2, 3 ICLK	section 11.
0008 C300h	ICU	Group 0 interrupt source register	GRP00	32	32	1 to 2PCLKB	2 ICLK	section 15.
0008 C304h	ICU	Group 1 interrupt source register	GRP01	32	32	1 to 2PCLKB	2 ICLK	
0008 C308h	ICU	Group 2 interrupt source register	GRP02	32	32	1 to 2PCLKB	2 ICLK	
0008 C30Ch	ICU	Group 3 interrupt source register	GRP03	32	32	1 to 2PCLKB	2 ICLK	
0008 C310h	ICU	Group 4 interrupt source register	GRP04	32	32	1 to 2PCLKB	2 ICLK	
0008 C314h	ICU	Group 5 interrupt source register	GRP05	32	32	1 to 2PCLKB	2 ICLK	
0008 C318h	ICU	Group 6 interrupt source register	GRP06	32	32	1 to 2PCLKB	2 ICLK	
0008 C330h	ICU	Group 12 interrupt source register	GRP12	32	32	1 to 2PCLKB	2 ICLK	
0008 C340h	ICU	Group 0 interrupt enable register	GEN00	32	32	1 to 2PCLKB	2 ICLK	
0008 C344h	ICU	Group 1 interrupt enable register	GEN01	32	32	1 to 2PCLKB	2 ICLK	
0008 C348h	ICU	Group 2 interrupt enable register	GEN02	32	32	1 to 2PCLKB	2 ICLK	
0008 C34Ch	ICU	Group 3 interrupt enable register	GEN03	32	32	1 to 2PCLKB	2 ICLK	
0008 C350h	ICU	Group 4 interrupt enable register	GEN04	32	32	1 to 2PCLKB	2 ICLK	
0008 C354h	ICU	Group 5 interrupt enable register	GEN05	32	32	1 to 2PCLKB	2 ICLK	
0008 C358h	ICU	Group 6 interrupt enable register	GEN06	32	32	1 to 2PCLKB	2 ICLK	
0008 C370h	ICU	Group 12 interrupt enable register	GEN12	32	32	1 to 2PCLKB	2 ICLK	
0008 C380h	ICU	Group 0 interrupt clear register	GCR00	32	32	1 to 2PCLKB	2 ICLK	
0008 C384h	ICU	Group 1 interrupt clear register	GCR01	32	32	1 to 2PCLKB	2 ICLK	
0008 C388h	ICU	Group 2 interrupt clear register	GCR02	32	32	1 to 2PCLKB	2 ICLK	
0008 C38Ch	ICU	Group 3 interrupt clear register	GCR03	32	32	1 to 2PCLKB	2 ICLK	
0008 C390h	ICU	Group 4 interrupt clear register	GCR04	32	32	1 to 2PCLKB	2 ICLK	
0008 C394h	ICU	Group 5 interrupt clear register	GCR05	32	32	1 to 2PCLKB	2 ICLK	
0008 C398h	ICU	Group 6 interrupt clear register	GCR06	32	32	1 to 2PCLKB	2 ICLK	
0008 C3B0h	ICU	Group 12 interrupt clear register	GCR12	32	32	1 to 2PCLKB	2 ICLK	
0008 C3C0h	ICU	Unit select register	SEL	32	32	1 to 2PCLKB	2 ICLK	
0008 C400h	RTC	64-Hz counter	R64CNT	8	8	2, 3 PCLKB	2 ICLK	section 28.
0008 C402h	RTC	Second counter	RSECCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 C404h	RTC	Minute counter	RMINCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 C406h	RTC	Hour counter	RHRCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 C408h	RTC	Day-of-week counter	RWKCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 C40Ah	RTC	Date counter	RDAYCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 C40Ch	RTC	Month counter	RMONCNT	8	8	2, 3 PCLKB	2 ICLK	
0008 C40Eh	RTC	Year counter	RYRCNT	16	16	2, 3 PCLKB	2 ICLK	
0008 C410h	RTC	Second alarm register	RSECAR	8	8	2, 3 PCLKB	2 ICLK	
0008 C412h	RTC	Minute alarm register	RMINAR	8	8	2, 3 PCLKB	2 ICLK	
0008 C414h	RTC	Hour alarm register	RHRAR	8	8	2, 3 PCLKB	2 ICLK	
0008 C416h	RTC	Day-of-week alarm register	RWKAR	8	8	2, 3 PCLKB	2 ICLK	
0008 C418h	RTC	Date alarm register	RDAYAR	8	8	2, 3 PCLKB	2 ICLK	
0008 C41Ah	RTC	Month alarm register	RMONAR	8	8	2, 3 PCLKB	2 ICLK	
0008 C41Ch	RTC	Year alarm register	RYRAR	16	16	2, 3 PCLKB	2 ICLK	
0008 C41Eh	RTC	Year alarm enable register	RYRAREN	8	8	2, 3 PCLKB	2 ICLK	
0008 C422h	RTC	RTC control register 1	RCR1	8	8	2, 3 PCLKB	2 ICLK	
0008 C424h	RTC	RTC control register 2	RCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 C426h	RTC	RTC control register 3	RCR3	8	8	2, 3 PCLKB	2 ICLK	
0008 C428h	RTC	RTC control register 4	RCR4	8	8	2, 3 PCLKB	2 ICLK	
0008 C42Ah	RTC	Frequency register H	RFRH	16	16	2, 3 PCLKB	2 ICLK	
0008 C42Ch	RTC	Frequency register L	RFRL	16	16	2, 3 PCLKB	2 ICLK	
0008 C42Eh	RTC	Time error adjustment register	RADJ	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (34/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0008 C440h	RTC	Time capture control register 0	RTCCR0	8	8	2, 3 PCLKB	2 ICLK	section 28.
0008 C442h	RTC	Time capture control register 1	RTCCR1	8	8	2, 3 PCLKB	2 ICLK	
0008 C444h	RTC	Time capture control register 2	RTCCR2	8	8	2, 3 PCLKB	2 ICLK	
0008 C452h	RTC	Second capture register 0	RSECCP0	8	8	2, 3 PCLKB	2 ICLK	
0008 C454h	RTC	Minute capture register 0	RMINCP0	8	8	2, 3 PCLKB	2 ICLK	
0008 C456h	RTC	Hour capture register 0	RHRCP0	8	8	2, 3 PCLKB	2 ICLK	
0008 C45Ah	RTC	Date capture register 0	RDAYCP0	8	8	2, 3 PCLKB	2 ICLK	
0008 C45Ch	RTC	Month capture register 0	RMONCP0	8	8	2, 3 PCLKB	2 ICLK	
0008 C462h	RTC	Second capture register 1	RSECCP1	8	8	2, 3 PCLKB	2 ICLK	
0008 C464h	RTC	Minute capture register 1	RMINCP1	8	8	2, 3 PCLKB	2 ICLK	
0008 C466h	RTC	Hour capture register 1	RHRCP1	8	8	2, 3 PCLKB	2 ICLK	
0008 C46Ah	RTC	Date capture register 1	RDAYCP1	8	8	2, 3 PCLKB	2 ICLK	
0008 C46Ch	RTC	Month capture register 1	RMONCP1	8	8	2, 3 PCLKB	2 ICLK	
0008 C472h	RTC	Second capture register 2	RSECCP2	8	8	2, 3 PCLKB	2 ICLK	
0008 C474h	RTC	Minute capture register 2	RMINCP2	8	8	2, 3 PCLKB	2 ICLK	
0008 C476h	RTC	Hour capture register 2	RHRCP2	8	8	2, 3 PCLKB	2 ICLK	
0008 C47Ah	RTC	Date capture register 2	RDAYCP2	8	8	2, 3 PCLKB	2 ICLK	
0008 C47Ch	RTC	Month capture register 2	RMONCP2	8	8	2, 3 PCLKB	2 ICLK	
0008 C500h	TEMPS	Temperature sensor control register	TSCR	8	8	2, 3 PCLKB	2 ICLK	section 43.
0008 C880h	SYSTEM	Count clock extended register 1	SCK1	8	8	2, 3 PCLKB	2 ICLK	section 10.
0008 C890h	SYSTEM	Count clock extended register 2	SCK2	8	8	2, 3 PCLKB	2 ICLK	
0009 0200h to 0009 03FFh	CAN0	Mailbox registers 0 to 31	MB0 to 31	128	8, 16, 32	2, 3 PCLKB	2 ICLK	section 36.
0009 0400h to 0009 041Fh	CAN0	Mask registerer 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 0420h	CAN0	FIFO received ID compare register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 0424h	CAN0	FIFO received ID compare register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 0428h	CAN0	Mask invalid register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 042Ch	CAN0	Mailbox interrupt enable register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 0820h to 0009 083Fh	CAN0	Message control registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	
0009 0840h	CAN0	Control register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	
0009 0842h	CAN0	Status register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	
0009 0844h	CAN0	Bit configuration register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 0848h	CAN0	Receive FIFO control register	RFCR	8	8	2, 3 PCLKB	2 ICLK	
0009 0849h	CAN0	Receive FIFO pointer control register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	
0009 084Ah	CAN0	Transmit FIFO control register	TFCR	8	8	2, 3 PCLKB	2 ICLK	
0009 084Bh	CAN0	Transmit FIFO pointer control register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	
0009 084Ch	CAN0	Error interrupt enable register	EIER	8	8	2, 3 PCLKB	2 ICLK	
0009 084Dh	CAN0	Error interrupt factor judge register	EIFR	8	8	2, 3 PCLKB	2 ICLK	
0009 084Eh	CAN0	Receive error count register	RECR	8	8	2, 3 PCLKB	2 ICLK	
0009 084Fh	CAN0	Transmit error count register	TECR	8	8	2, 3 PCLKB	2 ICLK	
0009 0850h	CAN0	Error code store register	ECSR	8	8	2, 3 PCLKB	2 ICLK	
0009 0851h	CAN0	Channel search support register	CSSR	8	8	2, 3 PCLKB	2 ICLK	
0009 0852h	CAN0	Mailbox search status register	MSSR	8	8	2, 3 PCLKB	2 ICLK	
0009 0853h	CAN0	Mailbox search mode register	MSMR	8	8	2, 3 PCLKB	2 ICLK	
0009 0854h	CAN0	Time stamp register	TSR	16	16	2, 3 PCLKB	2 ICLK	
0009 0856h	CAN0	Acceptance filter support register	AFSR	16	16	2, 3 PCLKB	2 ICLK	
0009 0858h	CAN0	Test control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0009 1200h to 0009 13FFh	CAN1	Mailbox registers 0 to 31	MB0 to 31	128	8, 16, 32	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (35/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0009 1400h to 0009 141Fh	CAN1	Mask register 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 36.
0009 1420h	CAN1	FIFO received ID compare register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 1424h	CAN1	FIFO received ID compare register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 1428h	CAN1	Mask invalid register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 142Ch	CAN1	Mailbox interrupt enable register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 1820h to 0009 183Fh	CAN1	Message control registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	
0009 1840h	CAN1	Control register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	
0009 1842h	CAN1	Status register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	
0009 1844h	CAN1	Bit configuration register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 1848h	CAN1	Receive FIFO control register	RFCR	8	8	2, 3 PCLKB	2 ICLK	
0009 1849h	CAN1	Receive FIFO pointer control register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	
0009 184Ah	CAN1	Transmit FIFO control register	TFCR	8	8	2, 3 PCLKB	2 ICLK	
0009 184Bh	CAN1	Transmit FIFO pointer control register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	
0009 184Ch	CAN1	Error interrupt enable register	EIER	8	8	2, 3 PCLKB	2 ICLK	
0009 184Dh	CAN1	Error interrupt factor judge register	EIFR	8	8	2, 3 PCLKB	2 ICLK	
0009 184Eh	CAN1	Receive error count register	RECR	8	8	2, 3 PCLKB	2 ICLK	
0009 184Fh	CAN1	Transmit error count register	TECR	8	8	2, 3 PCLKB	2 ICLK	
0009 1850h	CAN1	Error code store register	ECSR	8	8	2, 3 PCLKB	2 ICLK	
0009 1851h	CAN1	Channel search support register	CSSR	8	8	2, 3 PCLKB	2 ICLK	
0009 1852h	CAN1	Mailbox search status register	MSSR	8	8	2, 3 PCLKB	2 ICLK	
0009 1853h	CAN1	Mailbox search mode register	MSMR	8	8	2, 3 PCLKB	2 ICLK	
0009 1854h	CAN1	Time stamp register	TSR	16	8, 16	2, 3 PCLKB	2 ICLK	
0009 1856h	CAN1	Acceptance filter support register	AFSR	16	8, 16	2, 3 PCLKB	2 ICLK	
0009 1858h	CAN1	Test control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
0009 2200h to 0009 23FFh	CAN2	Mailbox registers 0 to 31	MB0 to 31	128	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 2400h to 0009 241Fh	CAN2	Mask register 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 2420h	CAN2	FIFO received ID compare register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 2424h	CAN2	FIFO received ID compare register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 2428h	CAN2	Mask invalid register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 242Ch	CAN2	Mailbox interrupt enable register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 2820h to 0009 283Fh	CAN2	Message control registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	
0009 2840h	CAN2	Control register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	
0009 2842h	CAN2	Status register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	
0009 2844h	CAN2	Bit configuration register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	
0009 2848h	CAN2	Receive FIFO control register	RFCR	8	8	2, 3 PCLKB	2 ICLK	
0009 2849h	CAN2	Receive FIFO pointer control register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	
0009 284Ah	CAN2	Transmit FIFO control register	TFCR	8	8	2, 3 PCLKB	2 ICLK	
0009 284Bh	CAN2	Transmit FIFO pointer control register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	
0009 284Ch	CAN2	Error interrupt enable register	EIER	8	8	2, 3 PCLKB	2 ICLK	
0009 284Dh	CAN2	Error interrupt factor judge register	EIFR	8	8	2, 3 PCLKB	2 ICLK	
0009 284Eh	CAN2	Receive error count register	RECR	8	8	2, 3 PCLKB	2 ICLK	
0009 284Fh	CAN2	Transmit error count register	TECR	8	8	2, 3 PCLKB	2 ICLK	
0009 2850h	CAN2	Error code store register	ECSR	8	8	2, 3 PCLKB	2 ICLK	
0009 2851h	CAN2	Channel search support register	CSSR	8	8	2, 3 PCLKB	2 ICLK	
0009 2852h	CAN2	Mailbox search status register	MSSR	8	8	2, 3 PCLKB	2 ICLK	
0009 2853h	CAN2	Mailbox search mode register	MSMR	8	8	2, 3 PCLKB	2 ICLK	

Table 5.1 List of I/O Registers (Address Order) (36/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
0009 2854h	CAN2	Time stamp register	TSR	16	16	2, 3 PCLKB	2 ICLK	section 36.
0009 2856h	CAN2	Acceptance filter support register	AFSR	16	16	2, 3 PCLKB	2 ICLK	
0009 2858h	CAN2	Test control register	TCR	8	8	2, 3 PCLKB	2 ICLK	
000A 0000h	USB0	System configuration control register	SYSCFG	16	16	3 to 4 PCLKB	2, 3 ICLK	section 33.
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)*6	
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)*6	
000A 0014h	USB0	CFIFO port register	CFIFO	16	8, 16	3 to 4 PCLKB	2, 3 ICLK	
000A 0018h	USB0	D0FIFO port register	D0FIFO	16	8, 16	3 to 4 PCLKB	2, 3 ICLK	
000A 001Ch	USB0	D1FIFO port register	D1FIFO	16	8, 16	3 to 4 PCLKB	2, 3 ICLK	
000A 0020h	USB0	CFIFO port select register	CFIFOSEL	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 0022h	USB0	CFIFO port control register	CFIFOCTR	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 0028h	USB0	D0FIFO port select register	D0FIFOSEL	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 002Ah	USB0	D0FIFO port control register	D0FIFOCTR	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 002Ch	USB0	D1FIFO port select register	D1FIFOSEL	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 002Eh	USB0	D1FIFO port control register	D1FIFOCTR	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 0030h	USB0	Interrupt status register 0	INTENB0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0032h	USB0	Interrupt status register 1	INTENB1	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0036h	USB0	BRDY interrupt status register	BRDYENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0038h	USB0	NRDY interrupt status register	NRDYENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 003Ah	USB0	BEMP interrupt status register	BEMPENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 003Ch	USB0	SOF output configuration register	SOFCFG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0040h	USB0	Interrupt status register 0	INTSTS0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	

Table 5.1 List of I/O Registers (Address Order) (37/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000A 0042h	USB0	Interrupt status register 1	INTSTS1	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	section 33.
000A 0046h	USB0	BRDY interrupt status register	BRDYSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0048h	USB0	NRDY interrupt status register	NRDYSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 004Ah	USB0	BEMP interrupt status register	BEMPSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 004Ch	USB0	Frame number register	FRMNUM	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 004Eh	USB0	Device state changing register	DVCHGR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0050h	USB0	USB address register	USBADDR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0054h	USB0	USB request type register	USBREQ	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0056h	USB0	USB request value register	USBVAL	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0058h	USB0	USB request index register	USBINDX	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 005Ah	USB0	USB request length register	USBLENG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 005Eh	USB0	DCP maximum packet size register	DCPMAXP	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	

Table 5.1 List of I/O Registers (Address Order) (38/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000A 0060h	USB0	DCP control register	DCPCTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	section 33.
000A 0064h	USB0	Pipe window select register	PIPESEL	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0068h	USB0	Pipe configuration register	PIPECFG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 006Ch	USB0	Pipe maximum packet size register	PIPEMAXP	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 006Eh	USB0	Pipe cycle control register	PIPEPERI	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0070h	USB0	Pipe 1 control register	PIPE1CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0072h	USB0	Pipe 2 control register	PIPE2CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0074h	USB0	Pipe 3 control register	PIPE3CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0076h	USB0	Pipe 4 control register	PIPE4CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0078h	USB0	Pipe 5 control register	PIPE5CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 007Ah	USB0	Pipe 6 control register	PIPE6CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 007Ch	USB0	Pipe 7 control register	PIPE7CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	

Table 5.1 List of I/O Registers (Address Order) (39/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000A 007Eh	USB0	Pipe 8 control register	PIPE8CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	section 33.
000A 0080h	USB0	Pipe 9 control register	PIPE9CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0090h	USB0	Pipe 1 transaction counter enable register	PIPE1TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0092h	USB0	Pipe 1 transaction counter register	PIPE1TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0094h	USB0	Pipe 2 transaction counter enable register	PIPE2TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0096h	USB0	Pipe 2 transaction counter register	PIPE2TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0098h	USB0	Pipe 3 transaction counter enable register	PIPE3TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 009Ah	USB0	Pipe 3 transaction counter register	PIPE3TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 009Ch	USB0	Pipe 4 transaction counter enable register	PIPE4TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 009Eh	USB0	Pipe 4 transaction counter register	PIPE4TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 00A0h	USB0	Pipe 5 transaction counter enable register	PIPE5TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 00A2h	USB0	Pipe 5 transaction counter register	PIPE5TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	

Table 5.1 List of I/O Registers (Address Order) (40/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000A 00D0h	USB0	Device address 0 configuration register	DEVADD0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	section 33.
000A 00D2h	USB0	Device address 1 configuration register	DEVADD1	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 00D4h	USB0	Device address 2 configuration register	DEVADD2	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 00D6h	USB0	Device address 3 configuration register	DEVADD3	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 00D8h	USB0	Device address 4 configuration register	DEVADD4	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 00DAh	USB0	Device address 5 configuration register	DEVADD5	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0200h	USB1	System configuration control register	SYSCFG	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 0204h	USB1	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)*6	
000A 0208h	USB1	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)*6	
000A 0214h	USB1	CFIFO port register	CFIFO	16	8, 16	3 to 4 PCLKB	2, 3 ICLK	
000A 0218h	USB1	D0FIFO port register	D0FIFO	16	8, 16	3 to 4 PCLKB	2, 3 ICLK	
000A 021Ch	USB1	D1FIFO port register	D1FIFO	16	8, 16	3 to 4 PCLKB	2, 3 ICLK	
000A 0220h	USB1	CFIFO port select register	CFIFOSEL	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 0222h	USB1	CFIFO port control register	CFIFOCTR	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 0228h	USB1	D0FIFO port select register	D0FIFOSEL	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 022Ah	USB1	D0FIFO port control register	D0FIFOCTR	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 022Ch	USB1	D1FIFO port select register	D1FIFOSEL	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 022Eh	USB1	D1FIFO port control register	D1FIFOCTR	16	16	3 to 4 PCLKB	2, 3 ICLK	
000A 0230h	USB1	Interrupt enable register 0	INTENB0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	

Table 5.1 List of I/O Registers (Address Order) (41/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000A 0236h	USB1	BRDY interrupt enable register	BRDYENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	section 33.
000A 0238h	USB1	NRDY interrupt enable register	NRDYENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 023Ah	USB1	BEMP interrupt enable register	BEMPENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 023Ch	USB1	SOF output configuration register	SOFCFG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0240h	USB1	Interrupt status register 0	INTSTS0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0246h	USB1	BRDY interrupt status register	BRDYSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0248h	USB1	NRDY interrupt status register	NRDYSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 024Ah	USB1	BEMP interrupt status register	BEMPSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 024Ch	USB1	Frame number register	FRMNUM	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 024Eh	USB1	Device state changing register	DVCHGR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0250h	USB1	USB address register	USBADDR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0254h	USB1	USB request type register	USBREQ	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	

Table 5.1 List of I/O Registers (Address Order) (42/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000A 0256h	USB1	USB request value register	USBVAL	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	section 33.
000A 0258h	USB1	USB request index register	USBINDX	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 025Ah	USB1	USB request length register	USBLENG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 025Ch	USB1	DCP configuration register	DCPCFG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 025Eh	USB1	DCP maximum packet size register	DCPMAXP	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 0260h	USB1	DCP control register	DCPCTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 0264h	USB1	Pipe window select register	PIPESEL	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 0268h	USB1	Pipe configuration register	PIPECFG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 026Ch	USB1	Pipe maximum packet size register	PIPEMAXP	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 026Eh	USB1	Pipe cycle control register	PIPEPERI	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 0270h	USB1	Pipe 1 control register	PIPE1CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	
000A 0272h	USB1	Pipe 2 control register	PIPE2CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/(frequency ratio of ICLK/PCLKB)*6	

Table 5.1 List of I/O Registers (Address Order) (43/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000A 0274h	USB1	Pipe 3 control register	PIPE3CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	section 33.
000A 0276h	USB1	Pipe 4 control register	PIPE4CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0278h	USB1	Pipe 5 control register	PIPE5CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 027Ah	USB1	Pipe 6 control register	PIPE6CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 027Ch	USB1	Pipe 7 control register	PIPE7CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 027Eh	USB1	Pipe 8 control register	PIPE8CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0280h	USB1	Pipe 9 control register	PIPE9CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0290h	USB1	Pipe 1 transaction counter enable register	PIPE1TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0292h	USB1	Pipe 1 transaction counter register	PIPE1TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0294h	USB1	Pipe 2 transaction counter enable register	PIPE2TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0296h	USB1	Pipe 2 transaction counter register	PIPE2TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0298h	USB1	Pipe 3 transaction counter enable register	PIPE3TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	

Table 5.1 List of I/O Registers (Address Order) (44/45)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000A 029Ah	USB1	Pipe 3 transaction counter register	PIPE3TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	section 33.
000A 029Ch	USB1	Pipe 4 transaction counter enable register	PIPE4TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 029Eh	USB1	Pipe 4 transaction counter register	PIPE4TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 02A0h	USB1	Pipe 5 transaction counter enable register	PIPE5TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 02A2h	USB1	Pipe 5 transaction counter register	PIPE5TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0400h	USB	Deep standby USB transceiver control/pin monitor register	DPUSR0R	32	32	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000A 0404h	USB	Deep standby USB suspend/resume interrupt register	DPUSR1R	32	32	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*6	
000C 0000h	EDMAC	EDMAC mode register	EDMR	32	32	5, 6PCLKA	—	section 32.
000C 0008h	EDMAC	EDMAC transmit request register	EDTRR	32	32	5, 6PCLKA	—	
000C 0010h	EDMAC	EDMAC receive request register	EDRRR	32	32	5, 6PCLKA	—	
000C 0018h	EDMAC	Transmit descriptor list start address register	TDLAR	32	32	5, 6PCLKA	—	
000C 0020h	EDMAC	Receive descriptor list start address register	RDLAR	32	32	5, 6PCLKA	—	
000C 0028h	EDMAC	ETHERC/EDMAC status register	EESR	32	32	5, 6PCLKA	—	
000C 0030h	EDMAC	ETHERC/EDMAC status interrupt permission register	EESIPR	32	32	5, 6PCLKA	—	
000C 0038h	EDMAC	Transmit/receive status copy enable register	TRSCER	32	32	5, 6PCLKA	—	
000C 0040h	EDMAC	Receive missed-frame counter register	RMFCR	32	32	5, 6PCLKA	—	
000C 0048h	EDMAC	Transmit FIFO threshold register	TFTR	32	32	5, 6PCLKA	—	
000C 0050h	EDMAC	FIFO depth register	FDR	32	32	5, 6PCLKA	—	
000C 0058h	EDMAC	Receiving method control register	RMCR	32	32	5, 6PCLKA	—	
000C 0064h	EDMAC	Transmit FIFO underrun counter	TFUCR	32	32	5, 6PCLKA	—	
000C 0068h	EDMAC	Receive FIFO overflow counter	RFOCR	32	32	5, 6PCLKA	—	
000C 006Ch	EDMAC	Independent output signal setting register	IOSR	32	32	5, 6PCLKA	—	
000C 0070h	EDMAC	Flow control start FIFO threshold setting register	FCFTR	32	32	5, 6PCLKA	—	
000C 0078h	EDMAC	Receive data padding insert register	RPADIR	32	32	5, 6PCLKA	—	
000C 007Ch	EDMAC	Transmit interrupt setting register	TRIMD	32	32	5, 6PCLKA	—	
000C 00C8h	EDMAC	Receive buffer write address register	RBWAR	32	32	5, 6PCLKA	—	
000C 00CCh	EDMAC	Receive descriptor fetch address register	RDFAR	32	32	5, 6PCLKA	—	
000C 00D4h	EDMAC	Transmit buffer read address register	TBRAR	32	32	5, 6PCLKA	—	

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

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Reference Section
						ICLK≥PCLK	ICLK<PCLK	
000C 00D8h	EDMAC	Transmit descriptor fetch address register	TDFAR	32	32	5, 6PCLKA	—	section 33.
000C 0100h	ETHERC	ETHERC mode register	ECMR	32	32	5, 6PCLKA	—	
000C 0108h	ETHERC	Receive frame length register	RFLR	32	32	5, 6PCLKA	—	
000C 0110h	ETHERC	ETHERC status register	ECSR	32	32	5, 6PCLKA	—	
000C 0118h	ETHERC	ETHERC interrupt permission register	ECSIPR	32	32	5, 6PCLKA	—	
000C 0120h	ETHERC	PHY interface register	PIR	32	32	5, 6PCLKA	—	
000C 0128h	ETHERC	PHY status register	PSR	32	32	5, 6PCLKA	—	
000C 0140h	ETHERC	Random number generation counter upper limit setting register	RDMLR	32	32	5, 6PCLKA	—	
000C 0150h	ETHERC	IPG register	IPGR	32	32	5, 6PCLKA	—	
000C 0154h	ETHERC	Automatic PAUSE frame register	APR	32	32	5, 6PCLKA	—	
000C 0158h	ETHERC	Manual PAUSE frame register	MPR	32	32	5, 6PCLKA	—	
000C 0160h	ETHERC	PAUSE Frame receive counter register	RFCF	32	32	5, 6PCLKA	—	
000C 0164h	ETHERC	Automatic PAUSE frame retransmit count register	TPAUSER	32	32	5, 6PCLKA	—	
000C 0168h	ETHERC	PAUSE frame retransmit counter register	TPAUSECR	32	32	5, 6PCLKA	—	
000C 016Ch	ETHERC	Broadcast frame receive count setting register	BCFRR	32	32	5, 6PCLKA	—	
000C 01C0h	ETHERC	MAC address high register	MAHR	32	32	5, 6PCLKA	—	
000C 01C8h	ETHERC	MAC address low register	MALR	32	32	5, 6PCLKA	—	
000C 01D0h	ETHERC	Transmit retry over counter register	TROCR	32	32	5, 6PCLKA	—	
000C 01D4h	ETHERC	Delayed collision detect counter register	CDCR	32	32	5, 6PCLKA	—	
000C 01D8h	ETHERC	Lost carrier counter register	LCCR	32	32	5, 6PCLKA	—	
000C 01DCh	ETHERC	Carrier not detect counter register	CNDCR	32	32	5, 6PCLKA	—	
000C 01E4h	ETHERC	CRC error frame receive counter register	CEFCR	32	32	5, 6PCLKA	—	
000C 01E8h	ETHERC	Frame receive error counter register	FRECR	32	32	5, 6PCLKA	—	
000C 01ECh	ETHERC	Too-short frame receive counter register	TSFRCR	32	32	5, 6PCLKA	—	
000C 01F0h	ETHERC	Too-long frame receive counter register	TLFRCR	32	32	5, 6PCLKA	—	
000C 01F4h	ETHERC	Residual-bit frame receive counter register	RFCR	32	32	5, 6PCLKA	—	
000C 01F8h	ETHERC	Multicast address frame receive counter register	MAFCR	32	32	5, 6PCLKA	—	
007F C402h	FLASH	Flash mode register	FMODR	8	8	2 to 4FCLK	2, 3 ICLK	section 45. section 46.
007F C410h	FLASH	Flash access status register	FASTAT	8	8	2 to 4FCLK	2, 3 ICLK	section 45. section 46.
007F C411h	FLASH	Flash access error interrupt enable register	FAEINT	8	8	2 to 4FCLK	2, 3 ICLK	section 45. section 46.
007F C412h	FLASH	Flash ready interrupt enable register	FRDYIE	8	8	2 to 4FCLK	2, 3 ICLK	section 45.
007F C440h	FLASH	E2 data flash read enable register 0	DFLRE0	16	16	2 to 4FCLK	2, 3 ICLK	section 46.
007F C442h	FLASH	E2 data flash read enable register 1	DFLRE1	16	16	2 to 4FCLK	2, 3 ICLK	
007F C450h	FLASH	E2 data flash programming/erasure enable register 0	DFLWE0	16	16	2 to 4FCLK	2, 3 ICLK	
007F C452h	FLASH	E2 data flash programming/erasure enable register 1	DFLWE1	16	16	2 to 4FCLK	2, 3 ICLK	
007F C454h	FLASH	FCU RAM enable register	FCURAME	16	16	2 to 4FCLK	2, 3 ICLK	section 45.
007F FFB0h	FLASH	Flash status register 0	FSTATR0	8	8	2 to 4FCLK	2, 3 ICLK	
007F FFB1h	FLASH	Flash status register 1	FSTATR1	8	8	2 to 4FCLK	2, 3 ICLK	
007F FFB2h	FLASH	Flash P/E mode entry register	FENTRYR	16	16	2 to 4FCLK	2, 3 ICLK	
007F FFB4h	FLASH	Flash protection register	FPROTR	16	16	2 to 4FCLK	2, 3 ICLK	
007F FFB6h	FLASH	Flash reset register	FRESETR	16	16	2 to 4FCLK	2, 3 ICLK	
007F FFBAh	FLASH	FCU command register	FCMDR	16	16	2 to 4FCLK	2, 3 ICLK	
007F FFC8h	FLASH	FCU processing switching register	FCPSR	16	16	2 to 4FCLK	2, 3 ICLK	
007F FFCAh	FLASH	E2 data flash blank check control register	DFLBCCNT	16	16	2 to 4FCLK	2, 3 ICLK	section 46.
007F FFCCh	FLASH	Flash P/E status registe	FPESTAT	16	16	2 to 4FCLK	2, 3 ICLK	section 45.
007F FFCEh	FLASH	E2 data flash blank check status register	DFLBCSTAT	16	16	2 to 4FCLK	2, 3 ICLK	section 46.
007F FFE8h	FLASH	Peripheral clock notification register	PCKAR	16	16	2 to 4FCLK	2, 3 ICLK	section 45.

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 000881ECh. When different output triggers are specified, the PPG0.NDRH addresses for pulse output groups 2 and 3 are 000881EEh and 000881ECh, 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 000881EDh. When different output triggers are specified, the PPG0.NDRL addresses for pulse output groups 0 and 1 are 000881EFh and 000881EDh, 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 000881FCh. When different output triggers are specified, the PPG1.NDRH addresses for pulse output groups 6 and 7 are 000881FEh and 000881FCh, 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 000881FDh. When different output triggers are specified, the PPG1.NDRL addresses for pulse output groups 4 and 5 are 000881FFh and 000881FDh, respectively.
Note 5. Odd addresses should not be accessed in 16-bit units. When accessing a register in 16-bit units, access the address of the TMR0 or TMR2 register. Table 26.4 lists register allocation for 16-bit access.
Note 6. When the register is accessed while the USB is operating, a delay may be generated in accessing.

6. Resets

6.1 Overview

There are nine types of resets: 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
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 the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), see section 8, Voltage Detection Circuit (LVD) and section 49, 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								
	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
Register related to the power-on reset circuit (RSTSR0.PORF)	○	—	—	—	—	—	—	—	—
Register related to the cold start/warm start determination flag (RSTSR1.CWSF)	—	○	—	—	—	—	—	—	—
Register related to the voltage monitor function 0 (RSTSR0.LVD0RF)	○	○	—	—	—	—	—	—	—
Registers related to the independent watchdog timer (RSTSR2.IWDTRF, IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCOCR)	○	○	○	—	—	—	—	○	—
Registers related to the watchdog timer (RSTSR2.WDTRF, WDTRR, WDTCR, WDTSR, WDTRCR)	○	○	○	○	—	—	—	○	—
Registers related to the voltage monitor function 1 (RSTSR0.LVD1RF, LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL)	○	○	○	○	○	—	—	—	—
(LVD1CR1, LVD1SR)	○	○	○	○	○	—	—	○	—
Registers related to the voltage monitor function 2 RSTSR0.LVD2RF, LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	○	○	○	○	○	○	—	—	—
(LVD2CR1, LVD2SR)	○	○	○	○	○	○	—	○	—
Register related to the low power-consumption function (RSTSR0.DPSRSTF)	○	○	○	○	○	○	○	—	—
Register related to the software reset (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.MOFXIN)	○	○	○	○	○	○	○	—	○
Pin state	○	○	○	○	○	○	○	—	○
Registers related to the low power-consumption function (DPSBYCR, DPSIER, DPSIFR, DPSIEGR, DPUSR0R, DPUSR1R)	○	○	○	○	○	○	○	—	○
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 reset. For details on the target bits, refer to section 28, Realtime Clock (RTCa).

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

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	0	0	0	0	0	0	0

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) *1
b1	LVD0RF	Voltage-Monitoring 0 Reset Detect Flag	0: Voltage-monitoring 0 reset not detected. 1: Voltage-monitoring 0 reset detected.	R(/W) *1
b2	LVD1RF	Voltage-Monitoring 1 Reset Detect Flag	0: Voltage-monitoring 1 reset not detected. 1: Voltage-monitoring 1 reset detected.	R(/W) *1
b3	LVD2RF	Voltage-Monitoring 2 Reset Detect Flag	0: Voltage-monitoring 2 reset not detected. 1: Voltage-monitoring 2 reset detected.	R(/W) *1
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) *1

Note 1. 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 resets shown in Table 6.2 occur.
- 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 VCC voltage has fallen below Vdet0.
[Setting condition]

- When Vdet0-level VCC voltage is detected.

[Clearing conditions]

- When resets listed in Table 6.2 occur.
- 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 VCC voltage has fallen below Vdet1.
[Setting condition]

- When Vdet1-level VCC voltage is detected.

[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 VCC voltage has fallen below Vdet2.
[Setting condition]

- When Vdet2-level VCC voltage is detected.

[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) occurs.
[Setting condition]

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

[Clearing conditions]

- When resets shown in Table 6.2 occur.
- 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

Note 1. The value after reset depends on the reset source.

Bit	Symbol	Bit Name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start	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.

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	0	0

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) *1
b1	WDTRF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected. 1: Watchdog timer reset detected.	R(/W) *1
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected. 1: Software reset detected.	R(/W) *1
b7 to b3	—	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.

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 WDRF 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

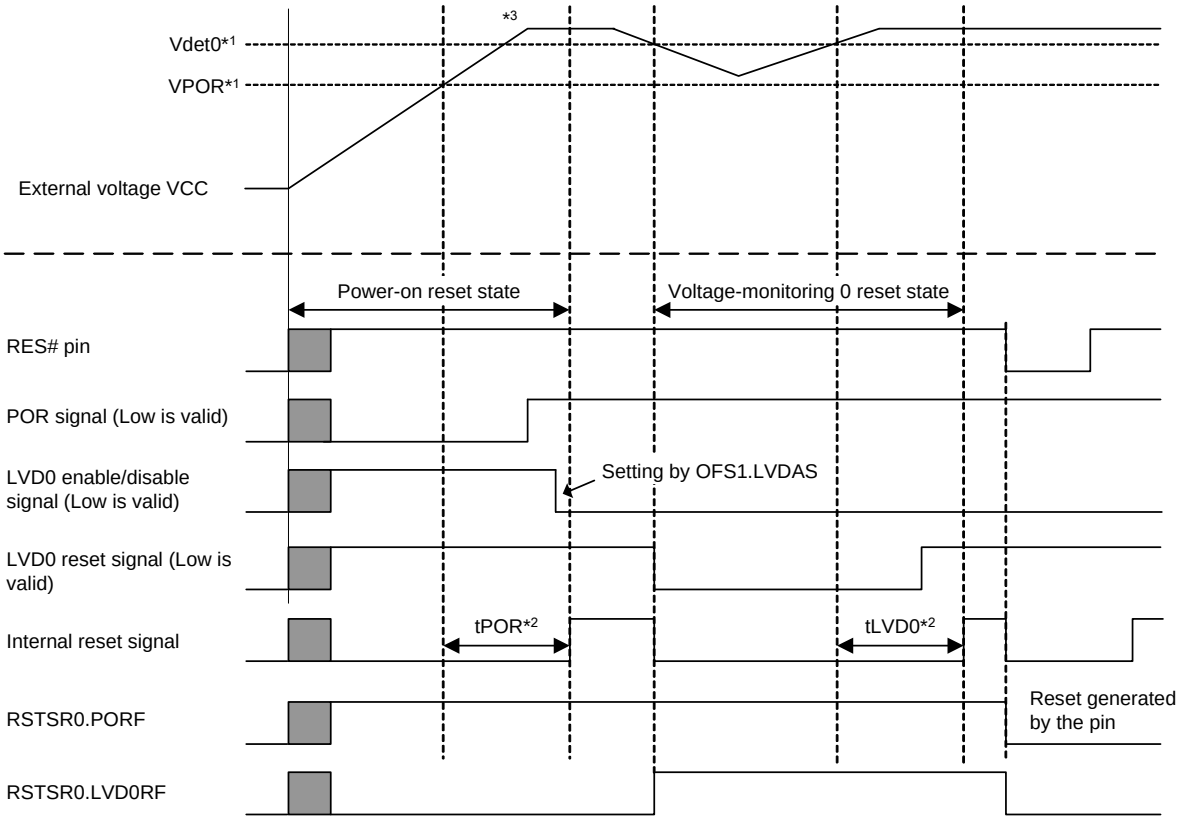
6.3 Operation

6.3.1 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. During operation, the RES# pin should be held low according to the specified reset pulse width.
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 49, 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. 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 a stabilization period of the external power supply and the LSI circuit. After a power-on reset has been generated, the PORF flag in RSTSR0 is set to 1. The PORF flag is initialized by a pin reset.
The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection 0 level selection (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.
Release from the voltage monitoring 0 reset state occurs when VCC rises above Vdet0 and the LVD0 reset time (+LVD0) elapses.
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 (LVD).



Note: • For details on the electrical characteristics, see section 49, 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 rise 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 RSTST0.LVD1RF flag is set to 1 and the voltage-detection circuit generates a voltage monitoring 1 reset if VCC falls to or below Vdet1.

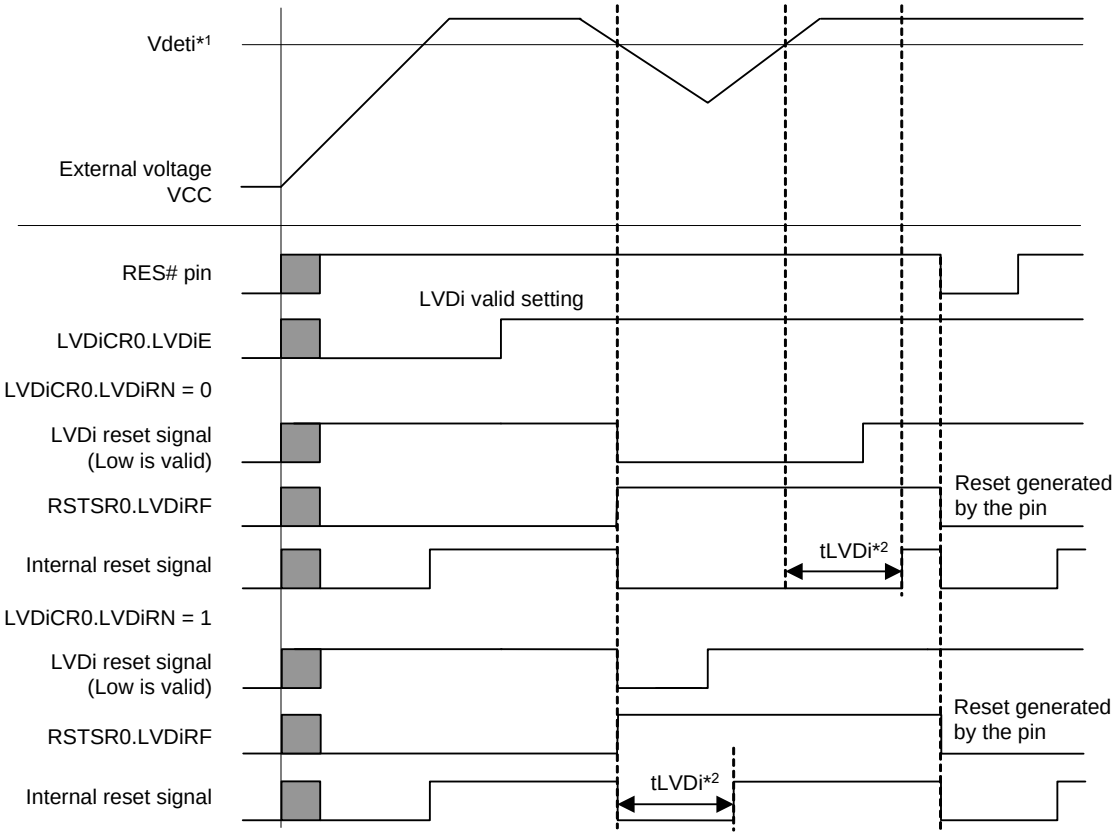
Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in voltage monitoring 2 circuit control register 0 (LVD2CR0), the RSTST0.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 LVD1 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 LVD1 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 LDV2CR0 register. Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection select register (LDVLVLR).

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 (LVD).



Note: • For details on the electrical characteristics, see section 49, 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 (return time after deep software standby mode cancelation) has elapsed. At the same time, deep software standby mode is also canceled.
When tDSBYWT (wait time after deep software standby mode cancelation) 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 watch dog 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 watch dog 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 30, 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 watch dog timer reset from the independent watchdog timer can be selected by settings in the WDT reset control register (WDTRCR) or option function select register 0 (OFS0).
When output of the independent watch dog 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 29, 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.

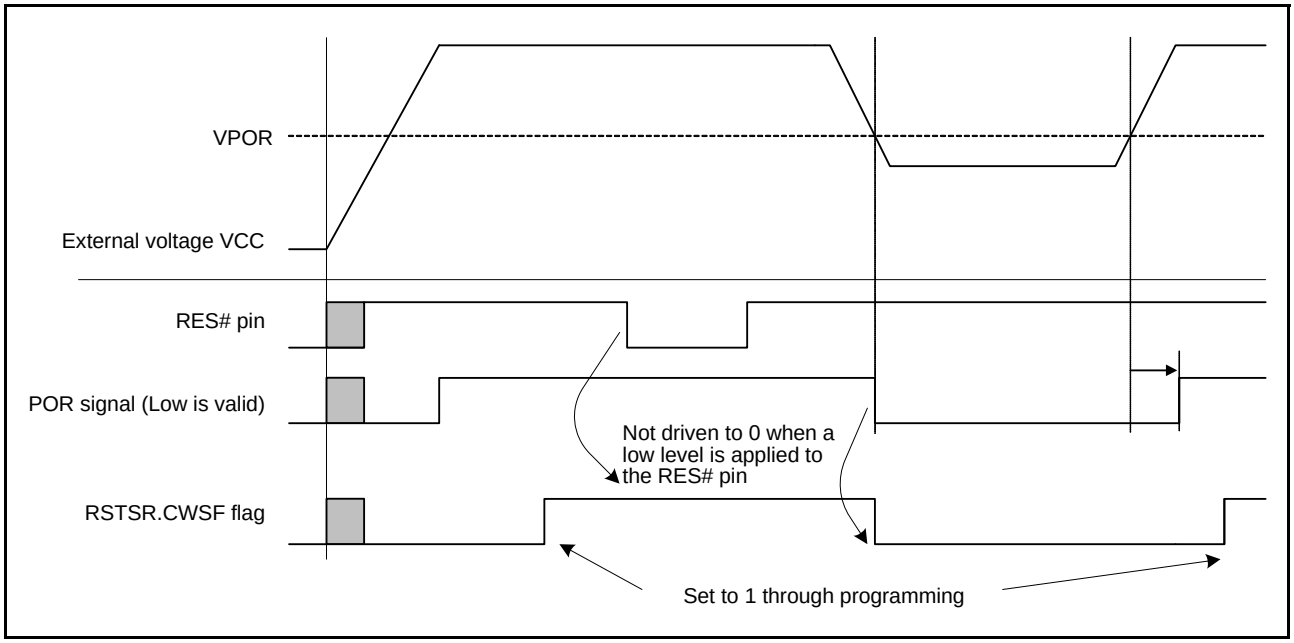


Figure 6.3 Example of Cold/Warm Start Determination Operation

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.

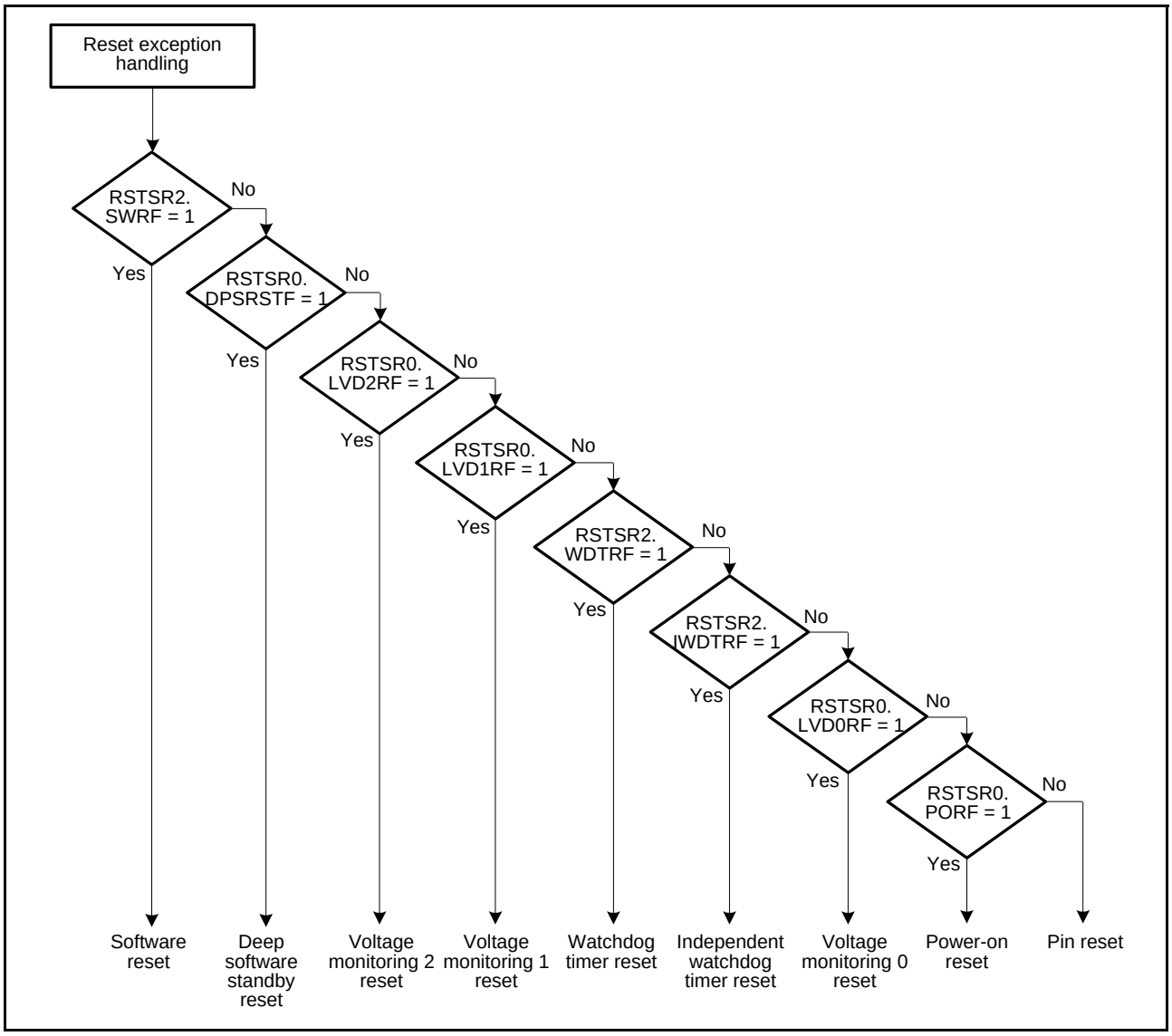


Figure 6.4 Example of Reset Generation Source Determination Flow

7. Option-Setting Memory

7.1 Overview

Option-setting memory refers to a set of registers that are provided for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the ROM.
Figure 7.1 shows the option-setting memory area.

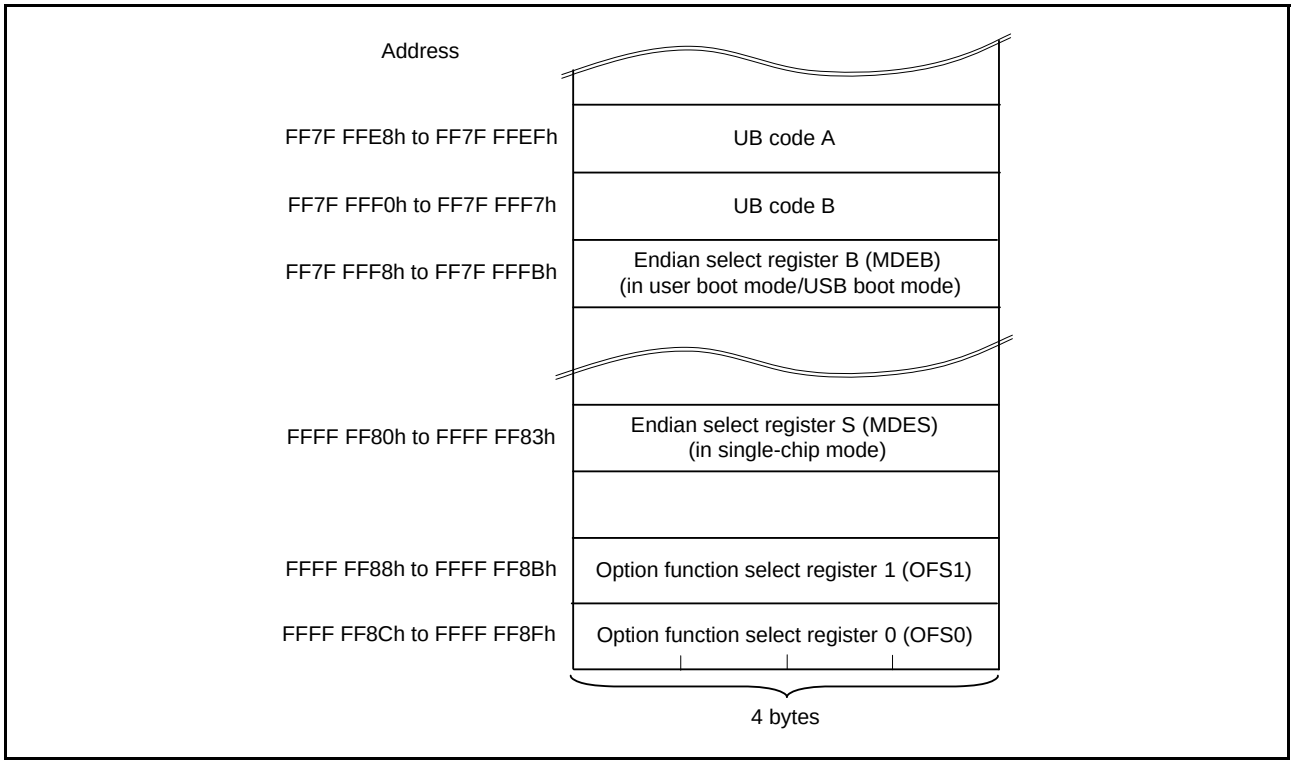


Figure 7.1 Option-Setting Memory Area

7.2 Register Descriptions

7.2.1 Option Function Select Register 0 (OFS0)

Address(es): FFFF FF8Ch

	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															

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns to 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]*1	IWDT Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: LOCO (Cycle period: 131 ms) 0 0 1 0: LOCO/16 (Cycle period: 2.10 s) 0 0 1 1: LOCO/32 (Cycle period: 4.19 s) 0 1 0 0: LOCO/64 (Cycle period: 8.39 s) 1 1 1 1: LOCO/128 (Cycle period: 16.8 s) 0 1 0 1: LOCO/256 (Cycle period: 33.6 s) 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 is enabled 1: Reset is enabled	R
b13	—	Reserved	When reading, this bit returns to 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, this bit returns to 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]*1	WDT Clock Frequency Division Ratio Select	b23 b20 0 0 0 1: PCLK/4 0 1 0 0: PCLK/64 1 1 1 1: PCLK/128 0 1 1 0: PCLK/512 0 1 1 1: PCLK/2048 1 0 0 0: PCLK/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 is enabled 1: Reset is enabled	R
b31 to b29	—	Reserved	When reading, this bit returns to the value written by the user. The write value should be 1.	R

Note 1. As for the blank product, the IWDTCKS[3:0] and WDTCKS[3:0] bits are 1111b (setting prohibited). The IWDTCKS[3:0] or WDTCKS[3:0] bits should be changed when the IWDT or WDT is activated in auto start mode by setting the IWDTSTRT or WDTSTRT bit to 0.

The OFS0 register is allocated in the ROM. Set this register at the same time as writing the program. After writing to the OFS0 register once, do not write to it again.
When erasing the block including the OFS0 register, the OFS0 register value becomes FFFF FFFFh.
The setting in the OFS0 register is ineffective in user boot mode and USB boot mode, and the value becomes 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 downcounter 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 LOCO clock cycles for 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 30, Independent Watchdog Timer (IWDTa).

IWDTCKS[3:0] Bits (IWDT 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 LOCO clock for 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 LOCO clock cycles for IWDT.
For details, see section 30, 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 30, 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 30, 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. Either an independent watchdog timer reset or a non-maskable interrupt request is selectable.
For details, refer to section 30, Independent Watchdog Timer (IWDTa).

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 30, 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 downcounter 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 PCLK 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 29, 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 PCLK. 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 PCLK cycles.
For details, see section 29, 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 29, 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 29, 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. Either a watchdog timer reset or a non-maskable interrupt request is selectable.
For details, refer to section 29, Watchdog Timer (WDTA).

7.2.2 Option Function Select Register 1 (OFS1)

Address(es): FFFF FF88h

	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
	—	—	—	—	—	—	—	HOCO EN	—	—	—	—	—	LVDAS	—	—
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.

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	When reading, this bit returns to the value written by the user. The write value should be 1.	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, this bit returns to 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, this bit returns to the value written by the user. The write value should be 1.	R

The OFS1 register is allocated in the ROM. Set this register at the same time as writing the program. After writing, do not write additions to this register.
When erasing the block including the OFS1 register, the setting in the OFS1 register is ineffective, and the OFS1 register value becomes FFFF FFFFh.
The setting in the OFS1 register is ineffective in user boot mode and USB boot mode, and the value becomes FFFF FFFFh.

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.

7.2.3 Endian Select Register B (MDEB), Endian Select Register S (MDES)

Address(es): FF7F FFF8h: MDEB (in user boot mode, or USB boot mode)
FFFF FF80h: MDES (in single-chip mode)

	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															

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

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, this bit returns to the value written by the user. The write value should be 1.	R

The MDEn (n = B, S) register selects the endian for the CPU. In user boot mode or USB boot mode, the endian select register B (MDEB) at address FF7F FFF8h is used to select the endian. In single ship mode, the endian select register S (MDES) at address FFFF FF80h is used.
MDEn is allocated in the ROM. Set the register at the same time as writing program. After writing to the register once, do not write to it again.
When erasing the block including the MDEn register, the MDEn register value becomes FFFF FFFFh.

MDE[2:0] Bits (Endian Select)

These bits select little endian or big endian for the CPU.
The endian is determined by the value at address FF7F FFF8h in the user boot area when operating in user boot mode or USB boot mode, and by the value at address FFFF FF80h in the user area when operating in single-chip mode.

7.3 UB Codes

UB codes A and B are required if user boot mode is to be employed. When the USB boot mode is used continuously, these codes should not be changed. The 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 55736572h and 426F6F74h.
- 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 PC7 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 other values for the code.
Figure 7.2 shows the structure of UB code A in memory. Set UB code A in 32-bit units.

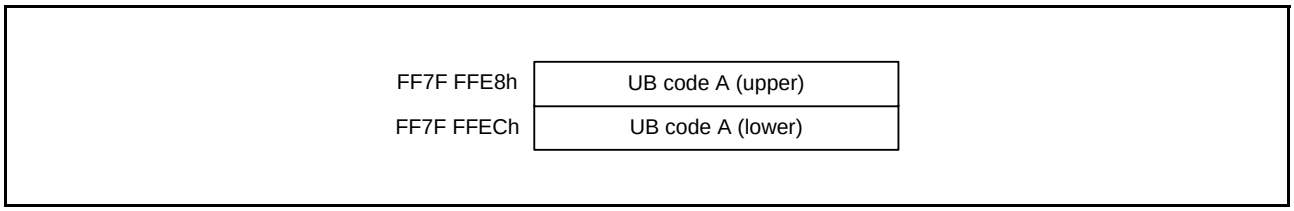


Figure 7.2 UB Code A Structure

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

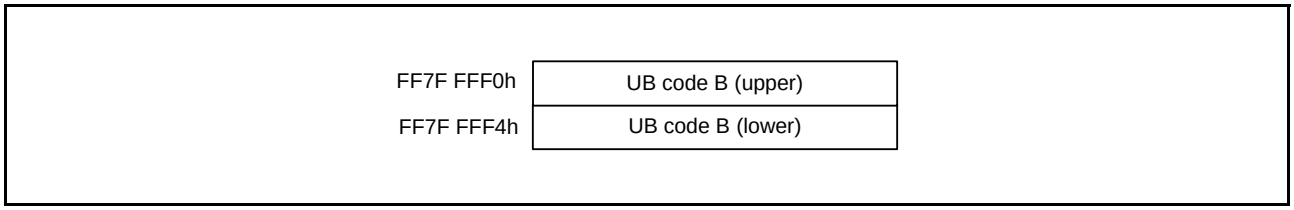


Figure 7.3 UB Code B Structure

7.4 Usage Note

7.4.1 Setting Example of Option-Setting Memory

Since the option-setting memory is allocated in the ROM, values cannot be written by executing instructions. Write appropriate values when writing the program. An example of the settings is shown below.

- To set ffff fff8h in the OFS0 register

```
.org    0ffff ff8ch
.lword  0ffffff8h
```

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

8. Voltage Detection Circuit (LVD)

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

8.1 Overview

In voltage detection 0, whether to enable or disable the reset of voltage monitoring 0 can be selected after the reset using the option function select register 1 (OFS1).
In voltage detection 1 and voltage detection 2, the detection voltage is set using the voltage detection level select register (LVDLVLR).
Reset of voltage monitoring 0, reset/interrupt of voltage monitoring 1, and reset/interrupt of 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	One level fixed	Specify voltage using LVDLVLR.LVD1LVLR[3:0] bits	Specify voltage 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	Non-maskable interrupt
			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: 1, 2, 4, 8)	1/n LOCO frequency × 2 (n: 1, 2, 4, 8)

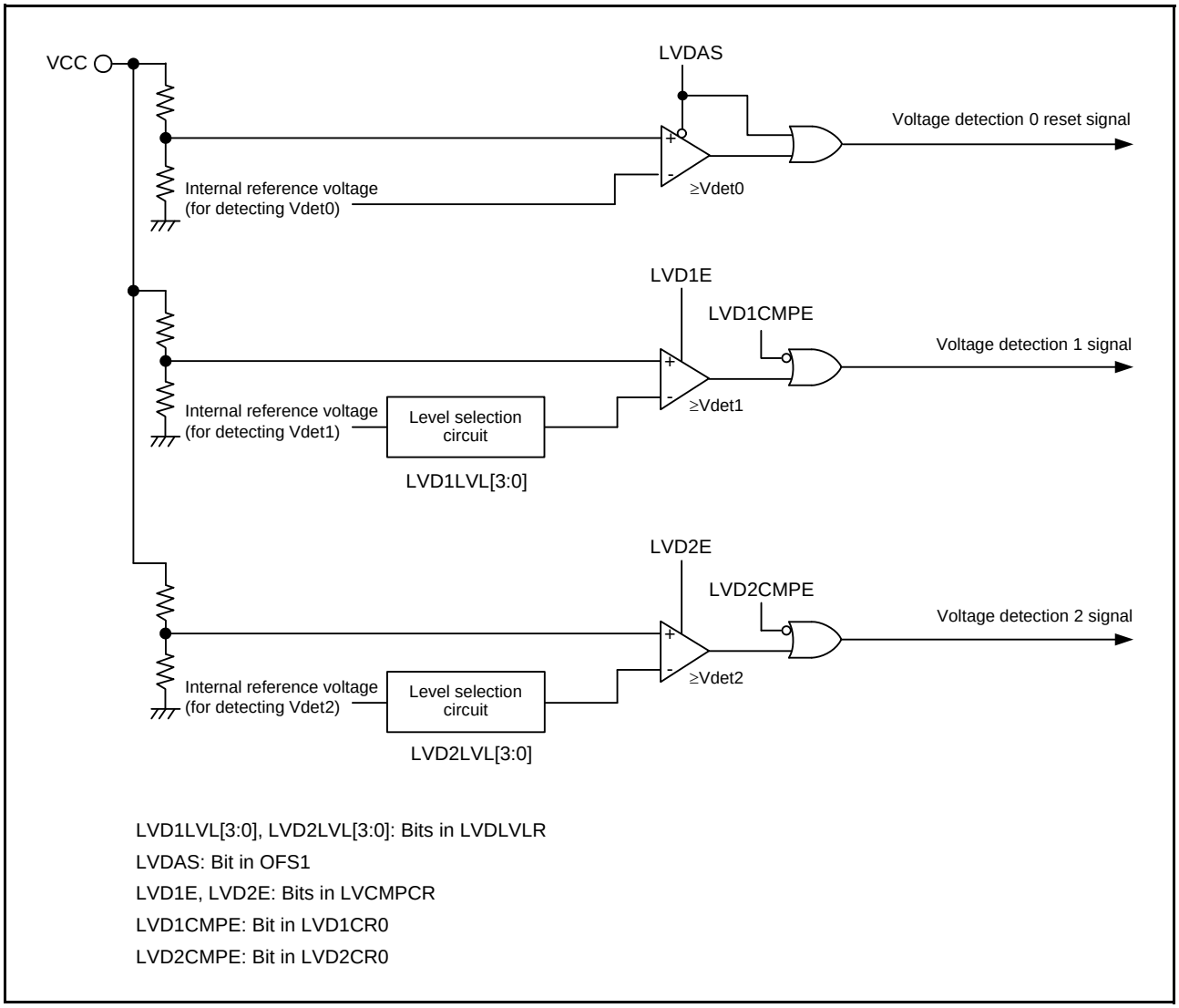


Figure 8.1 Block Diagram of Voltage Detection Circuit

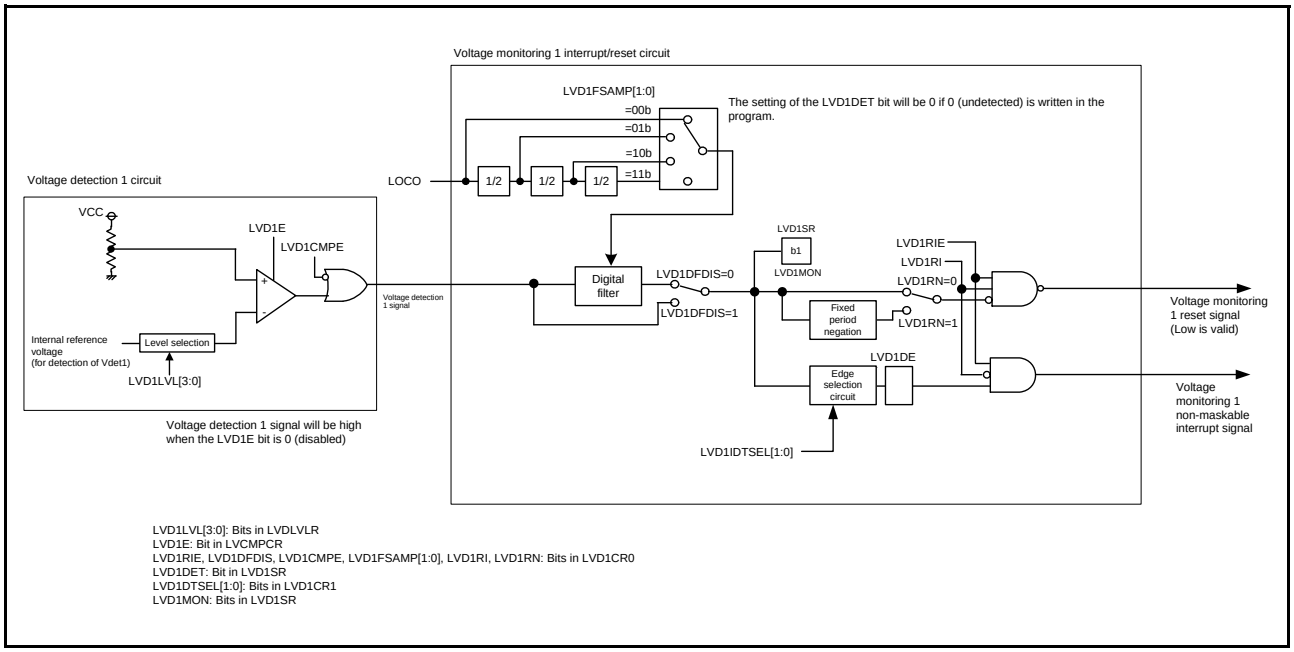


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

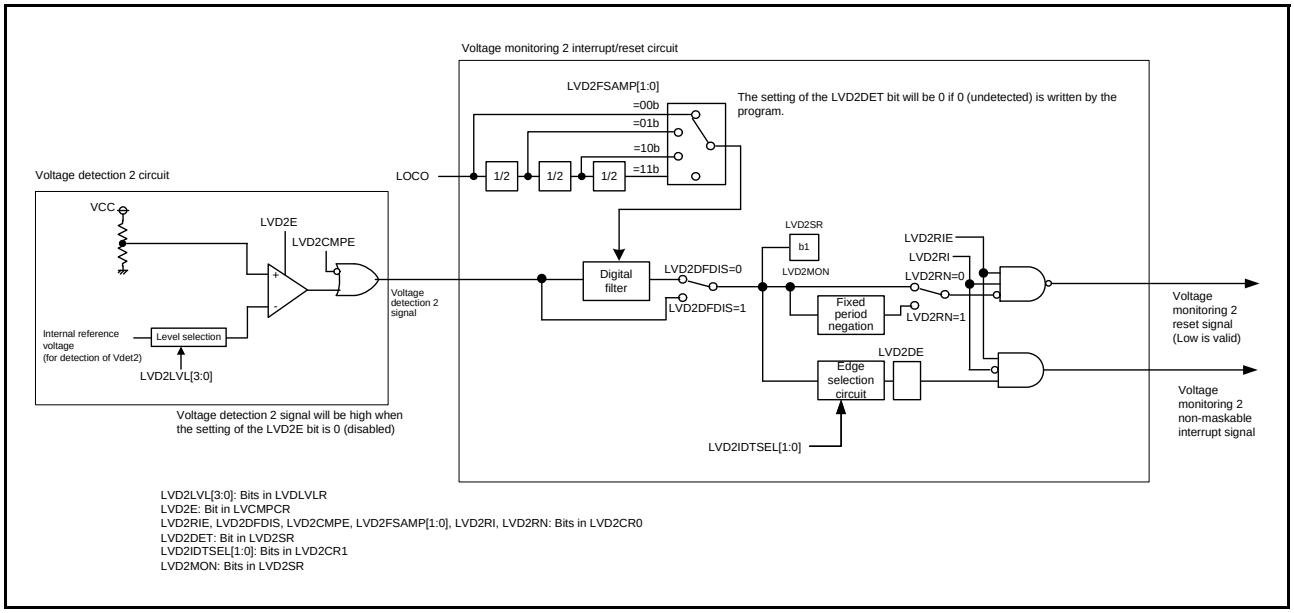


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

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
	—	—	—	—	—	—	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
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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).
When LVD1CR0 and LVD1CR1 have been modified, the LVD1DET flag may become 1. In this case, the LVD1DET flag should be set to 0 after LVD1CR0 and LVD1CR1 are modified.
The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) again 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
	—	—	—	—	—	—	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
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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).

When LVD2CR0 and LVD2CR1 have been modified, the LVD2DET flag may become 1. In this case, the LVD2DET flag should be set to 0 after LVD2CR0 and LVD2CR1 are modified.

The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE is set to 0 (disabled). LVD2CR0.LVD2RIE can be set to 1 (enabled) again 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	0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
b6	LVD2E	Voltage Detection 2 Enable	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

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	x	x	x	1	x	x	x

x: Undefined

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 b2 b1 b0 1 0 1 0: 2.95 V Do not set otherwise.	R/W
b7 to b4	LVD2LVL[3:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b7 b6 b5 b4 1 0 1 0: 2.95 V Do not set otherwise.	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.

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: 1/1 LOCO frequency 0 1: 1/2 LOCO frequency 1 0: 1/4 LOCO frequency 1 1: 1/8 LOCO frequency	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt enabled when Vdet1 is crossed 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 an LVD1 reset nor an LVD1 non-maskable 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 enabled) or when the LVD2CR0.LVD2RI bit is 1 (voltage monitoring 2 reset enabled), 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 enabled) and the LVD2CR0.LVD2RI bit to 0 (voltage monitoring 2 interrupt enabled).

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: 1/1 LOCO frequency 0 1: 1/2 LOCO frequency 1 0: 1/4 LOCO frequency 1 1: 1/8 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 an LVD2 reset nor an LVD2 non-maskable 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 enabled) or when the LVD1CR0.LVD1RI bit is 1 (voltage monitoring 1 reset enabled), 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 enabled) and the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt enabled).

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

After making the following settings, the LVD1SR.LVD1MON flag can be used to monitor the results of comparison by voltage monitor 1.

(1) Set the LVDLVLR.LVD1LVL[3:0] bits (detection voltage for voltage detection 1).
(2) Set the LVD1CR0.LVD1FSAMP[1:0] bits to select the sampling clock for the digital filter.
(3) Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by voltage monitor 1).
(4) Wait for at least one cycle of the LOCO.
(5) Clear the LVD1CR0.LVD1DFDIS bit (enabling the digital filter).
(6) Wait for at least 2n + 3 cycles of the LOCO (where n = 1, 2, 4, 8, and the sampling clock for the digital filter is the LOCO frequency-divided by n).
(7) Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).

Note: • When the digital filter is disabled, steps (2), (4), (5), and (6) are not required.

8.3.3 Monitoring Vdet2

After making the following settings, the LVD2SR.LVD2MON flag can be used to monitor the results of comparison by voltage monitor 2.

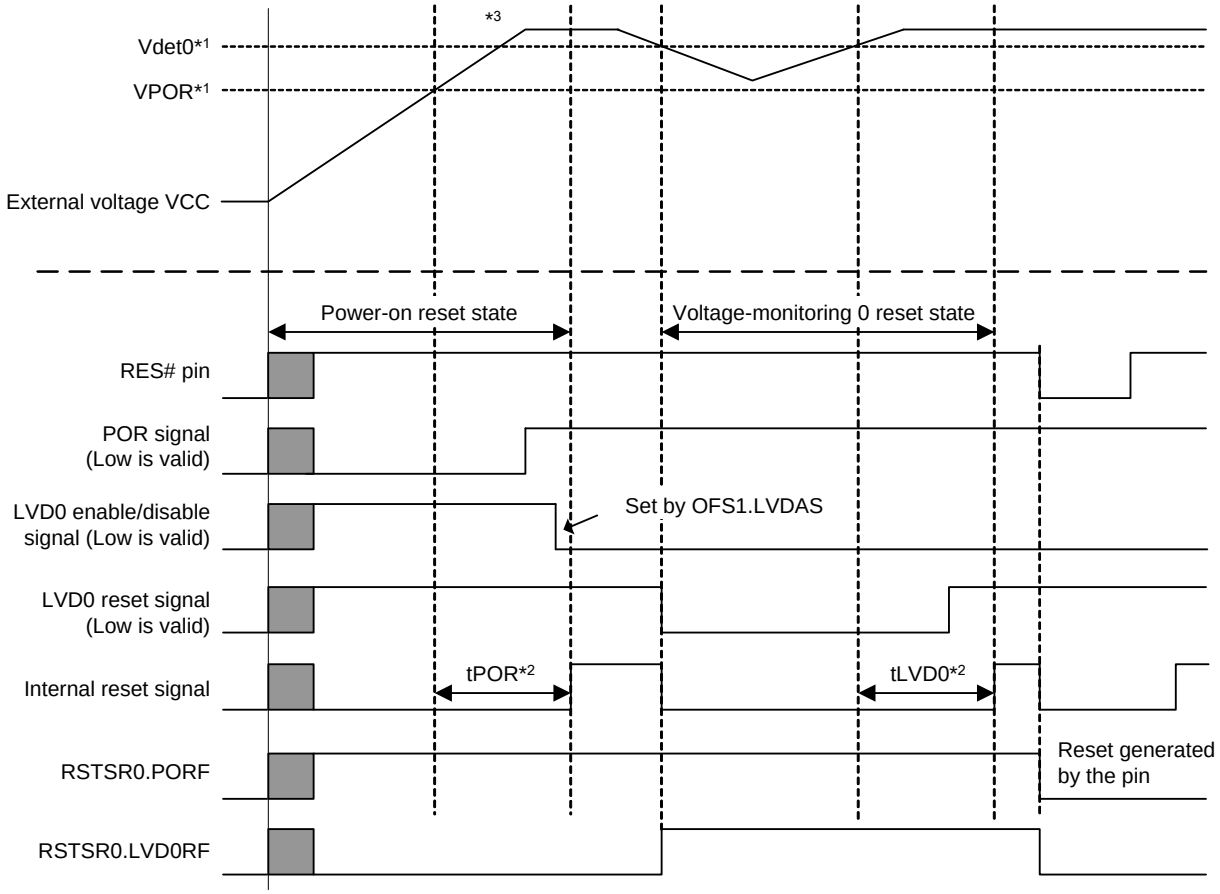
(1) Set the LVDLVLR.LVD2LVL[3:0] bits (detection voltage for voltage detection 2).
(2) Set the LVD2CR0.LVD2FSAMP[1:0] bits to select the sampling clock for the digital filter.
(3) Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by voltage monitor 2).
(4) Wait for at least one cycle of the LOCO.
(5) Clear the LVD2CR0.LVD2DFDIS bit (enabling the digital filter).
(6) Wait for at least 2n + 3 cycles of the LOCO (where n = 1, 2, 4, 8, and the sampling clock for the digital filter is the LOCO frequency-divided by n).
(7) Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled)

Note: • When the digital filter is disabled, steps (2), (4), (5), and (6) are not required.

8.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the voltage detection 0 circuit start bit (OFS1.LVDAS) 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 49, 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 or reset state.

Figure 8.4 Example of Voltage Monitoring 0 Reset Operation

8.5 Interrupt and Reset from Voltage Monitor 1

Table 8.2 lists the procedures for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring operates. Table 8.3 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, set the LVD1CR0.LVD1DFDIS bit to 1 (disabling the digital filter) if you intend to use the voltage monitor 1 circuit in software standby or deep software standby mode.

Table 8.2 Procedures for Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset so that Voltage Monitoring Operates

	When the Digital Filter is in Use		When the Digital Filter is Not in Use	
Step	**Voltage Monitor 1 Interrupt**	**Voltage Monitor 1 Reset**	**Voltage Monitor 1 Interrupt**	**Voltage Monitor 1 Reset**
1*1	Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.			
2*2	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.		Set the LVD1CR0.LVD1DFDIS bit to 1 (disabling the digital filter).	
3 *1, *2	Clear the LVD1CR0.LVD1RI bit to 0 (selecting the voltage monitor 1 interrupt).	• Set the LVD1CR0.LVD1RI bit to 1 (selecting the voltage monitor 1 reset). • Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.	Clear the LVD1CR0.LVD1RI bit to 0 (selecting the voltage monitor 1 interrupt).	• Set the LVD1CR0.LVD1RI bit to 1 (selecting the voltage monitor 1 reset). • Select the type of the reset negation by setting the LVD1CR0.LVD1RN bit.
4	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits.	—	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits.	—
5	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by voltage monitor 1).			
6	Wait for at least one cycle of the LOCO.		—	
7	Clear the LVD1CR0.LVD1DFDIS bit to 0 (enabling the digital filter).		—	
8	Wait for at least 2n + 3 cycles of the LOCO (where n = 1, 2, 4, 8, and the sampling clock for the digital filter is the LOCO frequency-divided by n).		— (waiting is not required)	
9	Clear the LVD1SR.LVD1DET flag to 0.	—	Clear the LVD1SR.LVD1DET flag to 0.	—
10	Set the LVD1CR0.LVD1RIE bit to 1 (enabling the voltage monitor 1 interrupt or reset).			
11*1	Set the LVCMPCR.LVD1E bit to 1 (enabling the voltage detection 1 circuit).			

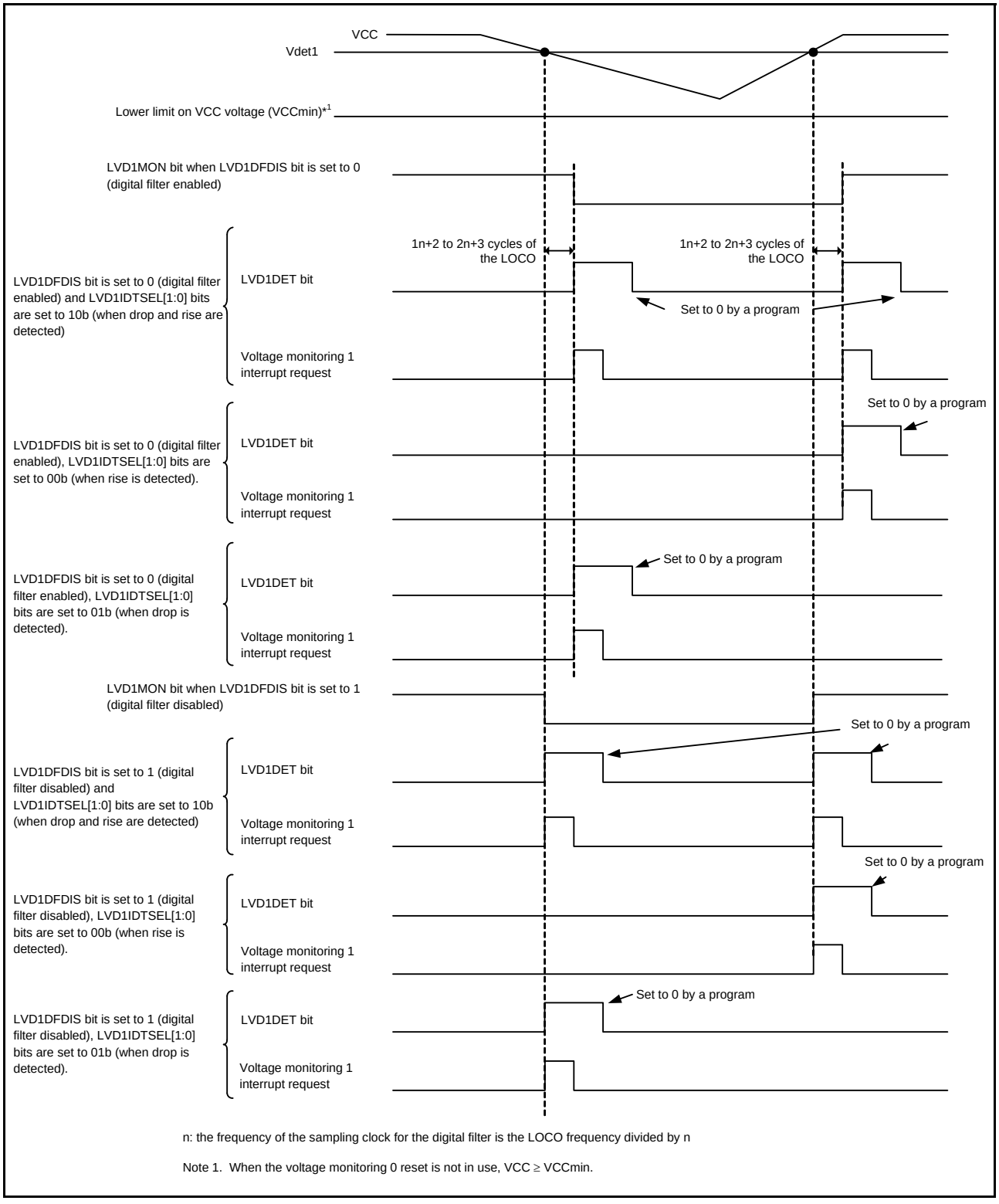
Note 1. Steps 1, 3, and 11 are not required if operation is with the setting to select the voltage monitor 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR0.LVD1FSAMP[1:0] and LVD1DFDIS bits or of the LVD1CR1.LVD1IDTSEL[1:0] bits, or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitor 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 11.
Note 2. Executing steps 2 and 3 at the same time (with a single instruction) creates no problems.

Table 8.3 Procedures for Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset to Stop Voltage Monitoring

Step	Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset to Stop Voltage Monitoring
1*1	Clear the LVCMPCR.LVD1E bit to 0 (disabling the voltage detection 1 circuit).
2*1	Wait for at least one cycle of the LOCO.
3	Clear the LVD1CR0.LVD1RIE bit to 0 (disabling the voltage monitor 1 interrupt or reset).
4	Clear the LVD1CR0.LVD1CMPE bit to 0 (disabling output of the results of comparison by voltage monitor 1).
5	Modify settings of bits related to the voltage detection circuit other than the LVCMPCR.LVD1E, LVD1CR0.LVD1CMPE, and LVD1RIE bits.

Note 1. Steps 1 and 2 are not required when operation is with the setting to select the voltage monitor 1 interrupt (LVD1CR0.LVD1RI = 0) and, after it is stopped, operation is to be restarted by simply changing the settings of the LVD1CR0.LVD1FSAMP[1:0] and LVD1DFDIS bits or the LVD1CR1.LVD1IDTSEL[1:0] bits, or when restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with

the setting to select the voltage monitor 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.



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 8.5 Example of Voltage Monitoring 1 Interrupt Operation

8.6 Interrupt and Reset from Voltage Monitor 2

Table 8.4 shows the procedures for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring operates.Table 8.5 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, set the LVD2CR0.LVD2DFDIS bit to 1 (disabling the digital filter) if you intend to use the voltage monitor 2 circuit in software standby or deep software standby mode.

Table 8.4 Procedures for Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset so that Voltage Monitoring Operates

	When the Digital Filter is in Use		When the Digital Filter is Not in Use	
Step	**Voltage Monitor 2 Interrupt**	**Voltage Monitor 2 Reset**	**Voltage Monitor 2 Interrupt**	**Voltage Monitor 2 Reset**
1*1	Specify the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.			
2*2	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.		Set the LVD2CR0.LVD2DFDIS bit to 1 (disabling the digital filter).	
3 *1,*2	Clear the LVD2CR0.LVD2RI bit to 0 (selecting the voltage monitor 2 interrupt).	• Set the LVD2CR0.LVD2RI bit to 1 (selecting the voltage monitor 2 reset). • Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.	Clear the LVD2CR0.LVD2RI bit to 0 (selecting the voltage monitor 2 interrupt).	• Set the LVD2CR0.LVD2RI bit to 1 (selecting the voltage monitor 2 reset). • Select the type of the reset negation by setting the LVD2CR0.LVD2RN bit.
4	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits.	—	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits.	—
5	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by voltage monitor 2).			
6	Wait for at least one cycle of the LOCO.		—	
7	Clear the LVD2CR0.LVD2DFDIS bit to 0 (enabling the digital filter).		—	
8	Wait for at least 2n + 3 cycles of the LOCO (where n = 1, 2, 4, 8, and the sampling clock for the digital filter is the LOCO frequency-divided by n).		— (waiting is not required)	
9	Clear the LVD2SR.LVD2DET flag to 0.	—	Clear the LVD2SR.LVD2DET flag to 0.	—
10	Set the LVD2CR0.LVD2RIE bit to 1 (enabling the voltage monitor 2 interrupt or reset).			
11*1	Set the LVCMPCR.LVD2E bit to 1 (enabling the voltage detection 2 circuit).			

Note 1. Steps 1, 3, and 11 are not required if operation is with the setting to select the voltage monitor 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR0.LVD2FSAMP[1:0] and LVD2DFDIS bits or of the LVD2CR1.LVD2IDTSEL[1:0] bits, or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitor 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 11.
Note 2. Executing steps 2 and 3 at the same time (with a single instruction) creates no problems.

Table 8.5 Procedures for Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset to Stop Voltage Monitoring

Step	Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset to Stop Voltage Monitoring
1*1	Clear the LVCMPCR.LVD2E bit to 0 (disabling the voltage detection 2 circuit).
2*1	Wait for at least one cycle of the LOCO.
3	Clear the LVD2CR0.LVD2RIE bit to 0 (disabling the voltage monitor 2 interrupt or reset).
4	Clear the LVD2CR0.LVD2CMPE bit to 0 (disabling output of the results of comparison by voltage monitor 2).
5	Modify settings of bits related to the voltage detection circuit other than the LVCMPCR.LVD2E, LVD2CR0.LVD2CMPE, and LVD2RIE bits.

Note 1. Steps 1 and 2 are not required when operation is with the setting to select the voltage monitor 2 interrupt (LVD2CR0.LVD2RI = 0) and, after it is stopped, operation is to be restarted by simply changing the settings of the LVD2CR0.LVD2FSAMP[1:0] and LVD2DFDIS bits or the LVD2CR1.LVD2IDTSEL[1:0] bits, or when restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitor 2 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.

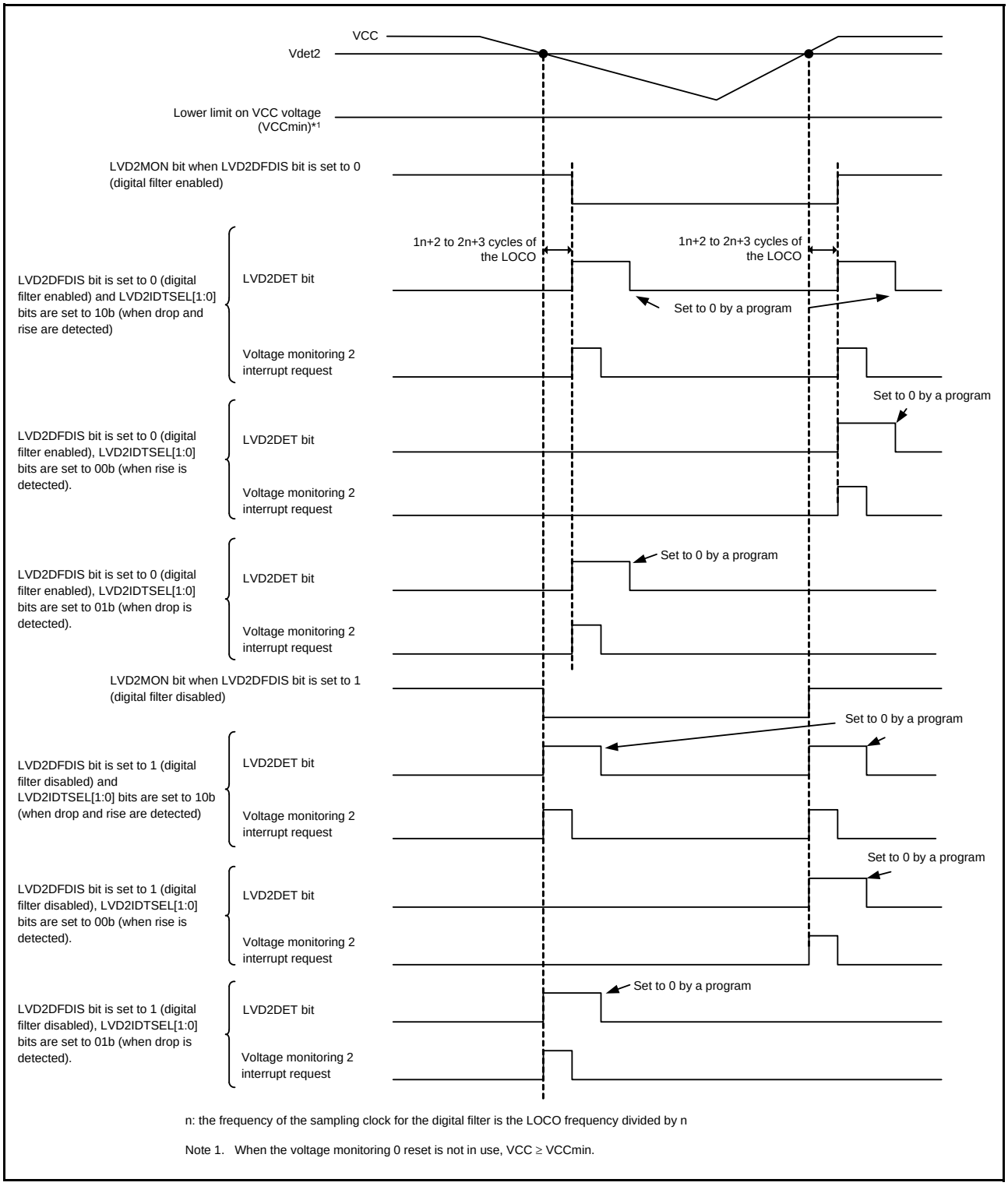


Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

9. Clock Generation Circuit

9.1 Overview

The RX63N/RX631 Group 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, ROM, and RAM. • Generates the peripheral module clock (PCLKA) to be supplied to the ETHERC and EDMAC.*1 • Generates the peripheral module clock (PCLKB) to be supplied to the peripheral module clocks other than the ETERC and EDMAC.*2 • Generates the FlashIF clock (FCLK) to be supplied to the FlashIF. • Generates the external bus clock (BCLK) to be supplied to the external bus. • Generates the SDRAM clock (SDCLK) to be supplied to SDRAM. • Generates the USB clock (UCLK) to be supplied to the USB. • Generates the CAN clock (CANCLK) to be supplied to the CAN. • Generates the IEBUS clock (IECLK) to be supplied to the IEBUS. • Generates the RTC-dedicated sub clock (RTCSCLK) to be supplied to the RTC. • Generates the RTC-dedicated main clock (RTCMCLK) to be supplied to the RTC. • Generates the IWDT-dedicated low-speed clock (IWDTCLK) to be supplied to the IWDT. • Generates the JTAG-dedicated clock (JTAGTCK) to be supplied to the JTAG.
Operating frequency	• ICLK: 100 MHz (max) • PCLKA: 100 MHz (max) • PCLKB: 50 MHz (max) • FCLK: 4 MHz to 50 MHz • BCLK: 100 MHz (max) • BCLK pin output: 50 MHz (max) • SDCLK pin output: 50 MHz (max) • UCLK: 48 MHz (max) • CANCLK: 20 MHz (max) • IECLK: 50 MHz (max) • RTCSCLK: 32.768 kHz • RTCMCLK: 4 MHz to 16 MHz • IWDTCLK: 125 kHz • JTAGTCK: 10 MHz (max)
Main clock oscillator	• Resonator frequency: 4 MHz to 16 MHz • External clock input frequency: 20 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 MTU2 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 circuit	• Input clock source: main clock • Input pulse frequency division ratio: Selectable from 1, 2, and 4 • Input frequency: 4 MHz to 16 MHz • Frequency multiplication ratio: Selectable from 8, 10, 12, 16, 20, 24, 25, and 50 • VCO oscillation frequency: 104 MHz to 200 MHz
High-speed on-chip oscillator (HOCO)	• Oscillation frequency: 50 MHz • HOCO power supply control
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 125 kHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 125 kHz
External clock input (TCK) for JTAG	Input clock frequency: 10 MHz (max)
Control of output on the BCLK pin	• BCLK clock output or high-level output is selectable • BCLK or BCLK/2 is selectable

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

Item	Specification
Control of output on the SDCLK pin	• SDCLK clock output or high-level output is selectable

Note 1. To use ETHERC, the following conditions must be met. 12.5 MHz ≤ PCLKA ≤ 100 MHz, PCLKA frequency = ICLK frequency
Note 2. PCLK indicates PCLKB.

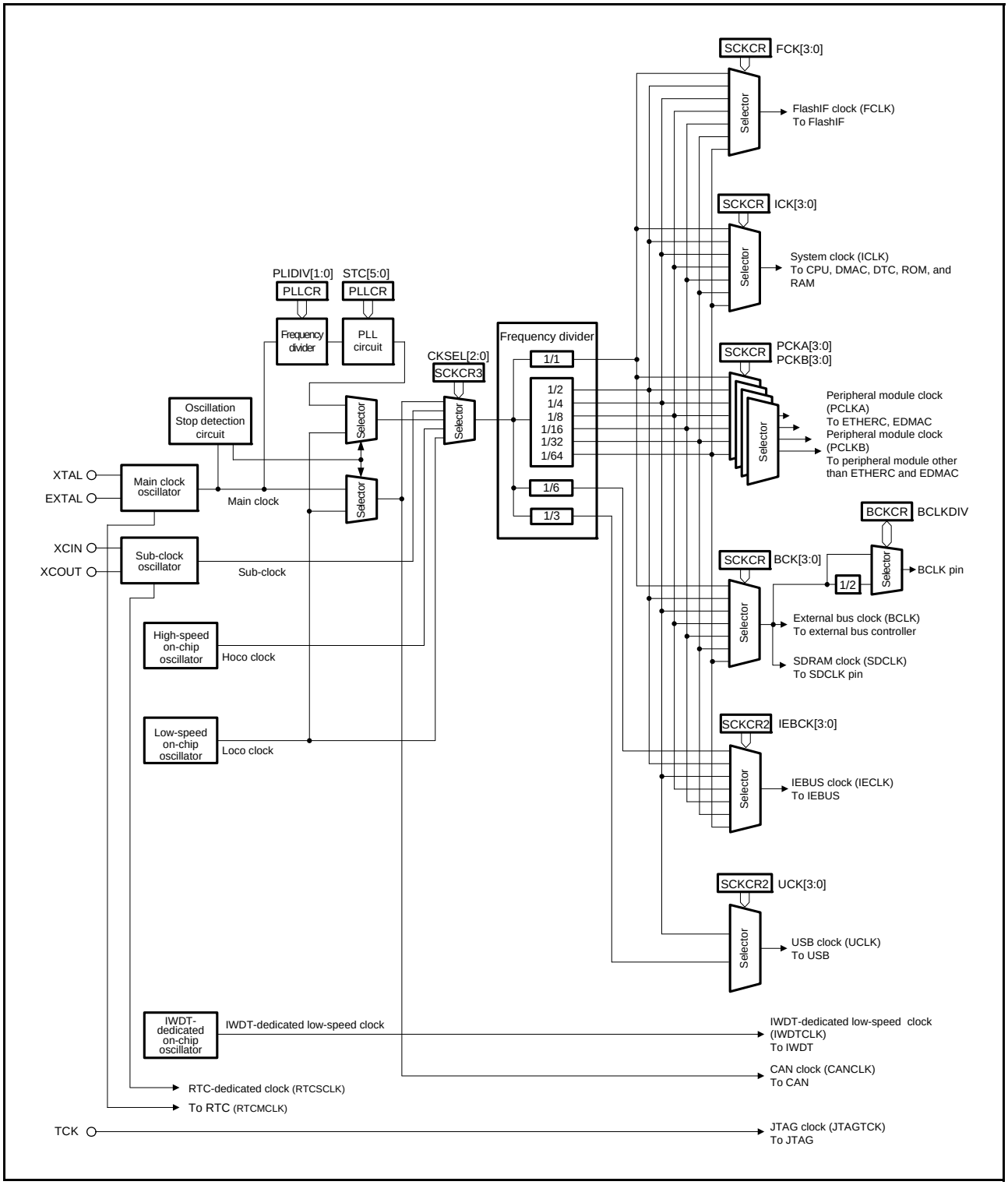


Figure 9.1 Block Diagram of Clock Generation Circuit

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 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]				—	—	—	—	—	—	—	—
Value after 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 should be set to 0001b.	R/W
b7 to b4	—	Reserved	These bits should be set to 0001b.	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: ×2 0 0 1 0: ×4 0 0 1 1: ×8 0 1 0 0: ×16 0 1 0 1: ×32 0 1 1 0: ×64 Settings other than those listed 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 to b20	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b22	PSTOP0	SDCLK Pin Output Control	0: SDCLK pin output is enabled. 1: SDCLK pin output is disabled. (Fixed high)	R/W
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

Bit	Symbol	Bit Name	Description	R/W
b31 to b28	FCK[3:0]	FlashIF 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 PLL is selected.
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 ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode) (i = 0 to 3)
- Time period from WAIT instruction issuance 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]				IEBCK[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
b3 to b0	IEBCK[3:0]	IEBUS Clock (IECLK) Select	b3 b0 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 1 1 0 0: ×1/6 Settings other than above are prohibited.	R/W
b7 to b4	UCK[3:0]	USB Clock (UCLK) Select	b7 b4 0 0 1 0: ×1/3 0 0 1 1: ×1/4 Settings other than above are prohibited when USB is in use. When USB is not in use, these bits are read as 0. The write value should be 0.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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 ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode) (i

= 0 to 3)

- Time period from WAIT instruction issuance 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 x1/3 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 ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode) (i = 0 to 3)
- Time period from WAIT instruction issuance 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 clock (PCLKA, PCLKB), FlashIF clock (FCLK), external bus clock (BCLK), IEBUS clock (IECLK), 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]						—	—	—	—	—	—	PLIDIV[1:0]	
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

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: ×1/4 1 1: Setting prohibited	R/W
b7 to b2	—	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 0 0 1 1 1: ×8 0 0 1 0 0 1: ×10 0 0 1 0 1 1: ×12 0 0 1 1 1 1: ×16 0 1 0 0 1 1: ×20 0 1 0 1 1 1: ×24 0 1 1 0 0 0: ×25 1 1 0 0 0 1: ×50 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

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 4 MHz to 16 MHz.

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 VCO oscillation frequency for the PLL (104 MHz to 200 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.

After the setting of the PLLEN bit has been changed so that the PLL operates, only start using the PLL after the PLL clock oscillation stabilization waiting time (tPLLWT1 or tPLLWT2) has elapsed.

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.

- When restarting the PLL after it has been stopped, allow at least five cycles of the PLL clock as an interval over which it is still stopped.
- Ensure that oscillation by the PLL is stable when making the setting to stop the PLL.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the PLL 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 PLL, wait for at least two cycles of the PLL clock 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 selection 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 ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode) (i = 0 to 3)
- Time period from WAIT instruction issuance 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.

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 deep software standby mode.

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 a crystal oscillator is connected to supply the main clock signal, after changing the MOSTP bit or MOFCR.MOFXIN bit so that the main clock oscillator operates, only use the main clock after the main clock oscillation stabilization waiting time (crystal; tMAINOSCWT) has elapsed.

When an external clock is connected to supply the main clock signal, after changing the MOSTP bit or MOFCR.MOFXIN bit so that the main clock oscillator operates, only use the main clock after the EXTAL external clock input waiting time (tEXWT) has elapsed.

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 PLL operation.

- When restarting the main clock after it has been stopped, allow at least five cycles of the main clock as an interval over which it is still stopped.
- Ensure that oscillation by the main clock oscillator is stable when making the setting to stop the main clock oscillator.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the main clock oscillator 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 main clock oscillator, wait for at least two cycles of the main clock before executing the WAIT instruction.

Writing of 1 to the MOSTP bit (stopping the main clock oscillator) is prohibited 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]).

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).

After the setting of the SOSTP bit or the RCR3.RTCEN bit has been changed so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization waiting time (tSUBOSCWT) has elapsed. That is, a fixed time for stabilization is required after the setting for sub-clock oscillator operation. 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 sub-clock oscillator after it has been stopped, allow at least five cycles of the sub-clock as an interval over which it is still stopped.
- Ensure that oscillation by the sub-clock oscillator is stable when making the setting to stop the sub-clock oscillator.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the sub-clock oscillator 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 sub-clock oscillator, wait for at least two cycles of the sub-clock oscillator 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 low-speed clock oscillation stabilization waiting time (tLOCOWT) has elapsed.

That is, a fixed time for stabilization of oscillation is required after the setting for LOCO operation. 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).

9.2.10 IWDT-Dedicated On-Chip Clock 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	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 so that the IWDT-dedicated on-chip oscillator operates, supply of the clock within the LSI only starts after a time of waiting for stabilization of the LOCO (tLOCOWT) has elapsed. If the IWDT dedicated low-speed clock is to be used, only start using the oscillator after this waiting time (tLOCOWT) has elapsed.

Ensure that oscillation by the IWDT-dedicated on-chip oscillator is stable before executing a WAIT instruction to place the chip on software standby or deep software standby.

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

Note 1. The HCSTP bit value after a reset is 0 when the HOCO oscillation enable bit in the option function select register 1 (OFS1.HOCOEN) is 0. The HCSTP bit value after a reset is 1 when the OFS1.HOCOEN bit is 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

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.

After the setting of the HCSTP bit has been changed so that the HOCO operates, supply of the clock within the LSI only starts after time corresponding to the HOCO clock oscillation stabilization waiting time 2 (tHOCOWT2) has elapsed. Only start using the oscillator after this waiting time (tHOCOWT2) has elapsed. That is, a fixed time for stabilization of oscillation is required after the setting for HOCO operation. 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 HOCO after it has been stopped, allow at least five cycles of the HOCO as an interval over which it is still stopped.
- Ensure that oscillation by the HOCO is stable when making the setting to stop the HOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the HOCO 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 HOCO, wait for at least two cycles of the HOCO 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]).

Writing of 0 to the HCSTP bit (making the HOCO operate) is prohibited when the setting of the operating-power control mode selection bits in the operating-power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 2.

9.2.12 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

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 cleared 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.

9.2.13 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: The main clock oscillation stop has not been detected. 1: The main clock oscillation stop 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.

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 bit is not cleared to 0 even though the main clock oscillation is restarted. The OSTDF bit is cleared 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 bit is cleared to 0 while the main clock oscillation is stopped, the OSTDF bit becomes 0 and then returns to 1.

When 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 bit cannot be modified to 0. The OSTDF bit 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.

9.2.14 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The EXTAL/XTAL pin is also used as a port. In the initial setting state, the pin is set as a port. The EXTAL/XTAL signal can be used as the RTC clock source even in deep software standby mode.

MOFXIN Bit (Main Clock Oscillator Forced Oscillation)

This bit controls forced oscillation of the main clock oscillator.

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).

9.2.15 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).

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 49, 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 selection 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 resonator.

A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, 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 resonator should be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.

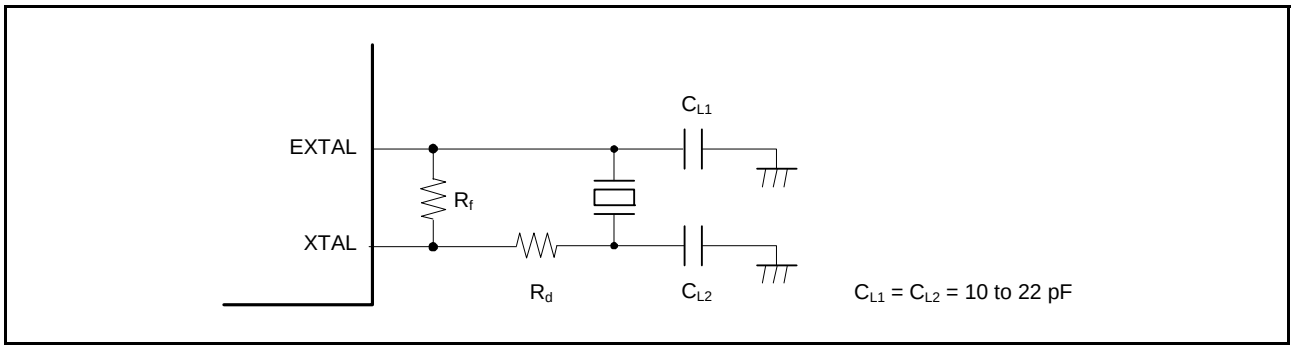


Figure 9.2 Example of Crystal Resonator Connection

Table 9.3 Damping Resistance (Reference Values)

Frequency (MHz)	4	8	12	16
Rd (Ω)	380	200	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.

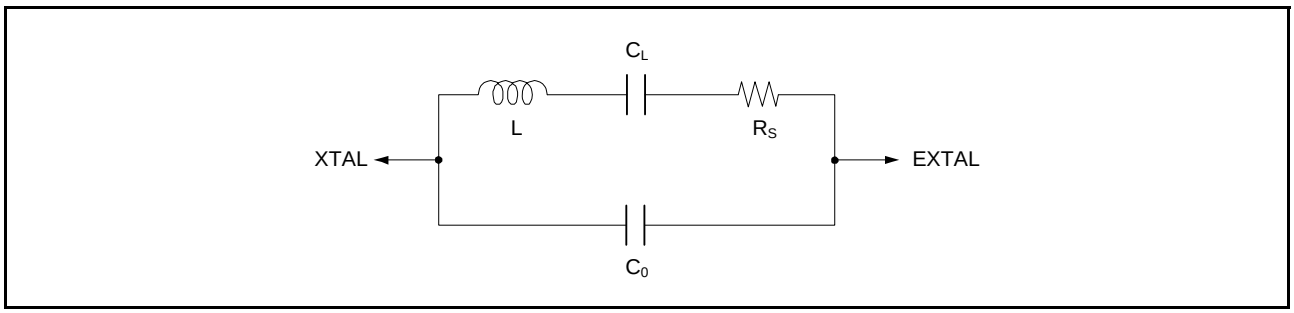


Figure 9.3 Equivalent Circuit of Crystal Resonator

Table 9.4 Crystal Resonator Characteristics (Reference Values)

Frequency (MHz)	4	8	12	16
R_S max (Ω)	160	80	60	40
C_0 max (pF)		7		

9.3.2 External Clock Input

Figure 9.4 shows examples of connection of external clock input. To leave the XTAL pin open, make the parasitic capacitance less than 5 pF.

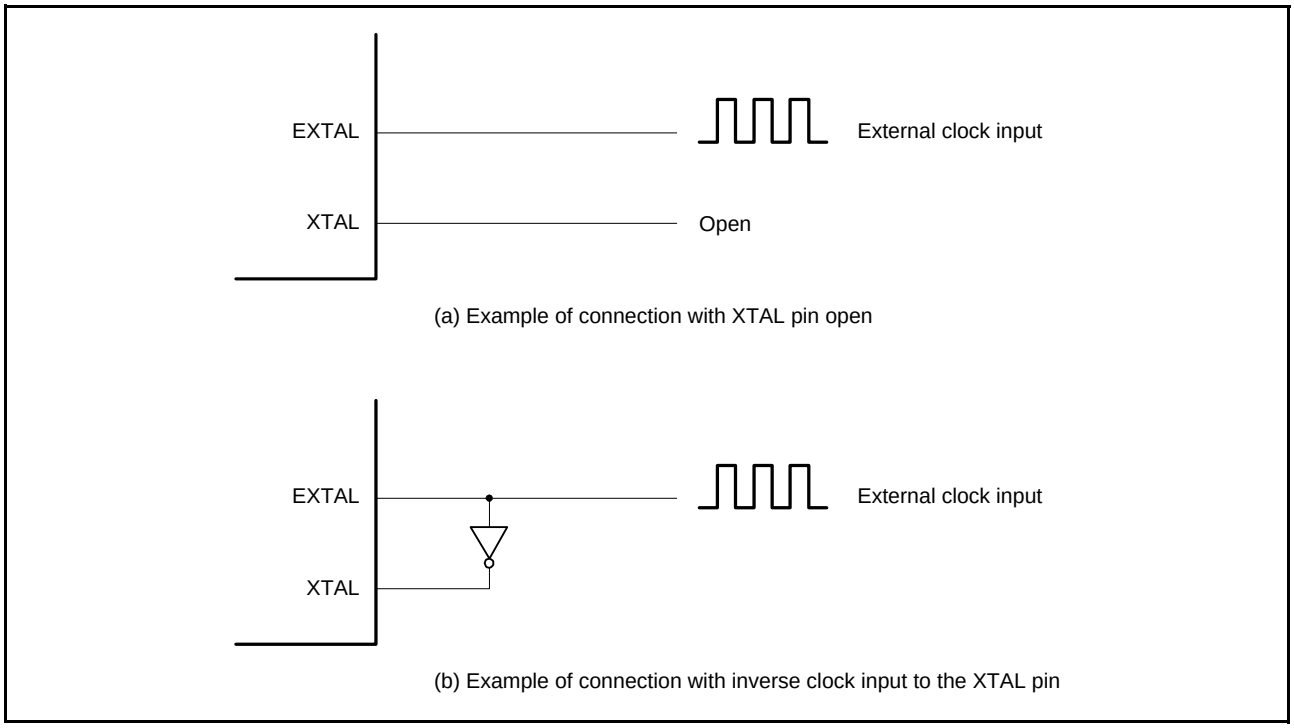


Figure 9.4 Examples of External Clock Input

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 forcible oscillation bit (MOFCR.MOFXIN) is 1 (forcing the main clock oscillator to run).

9.4 Sub-Clock Oscillator

The only way of supplying the clock signal to the sub-clock oscillator is connecting a crystal oscillator.

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 drive capability, 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.

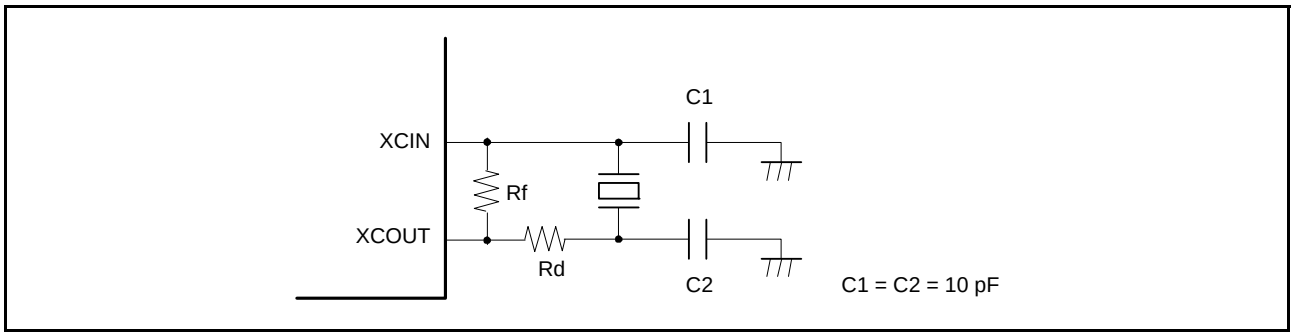


Figure 9.5 Connection Example of 32.768-kHz Crystal Resonator

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.

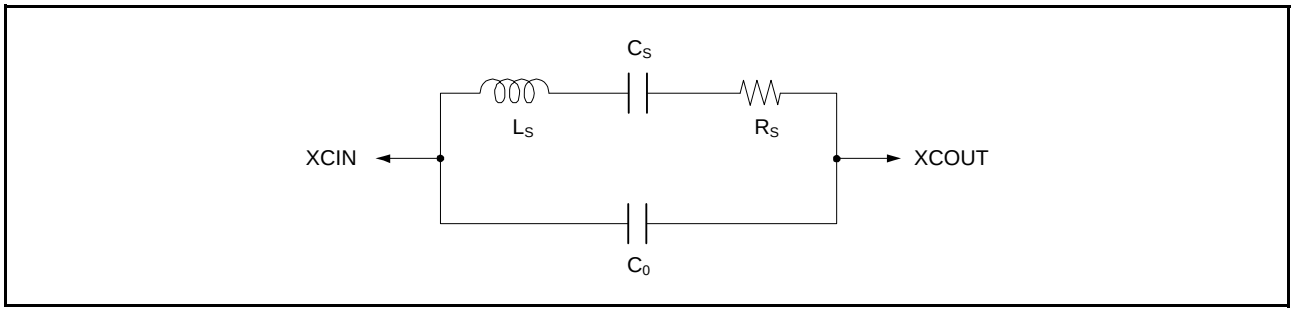


Figure 9.6 Equivalent Circuit for 32.768-kHz Crystal Resonator

Table 9.5 Crystal Resonator Characteristics (Reference Values)

Frequency (kHz)	32.768
R_S max (kΩ)	35
C_0 max (pF)	2.0

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 and leave the XCOUT pin open as shown in Figure 9.7. 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 (prohibiting input from the sub-clock oscillator).

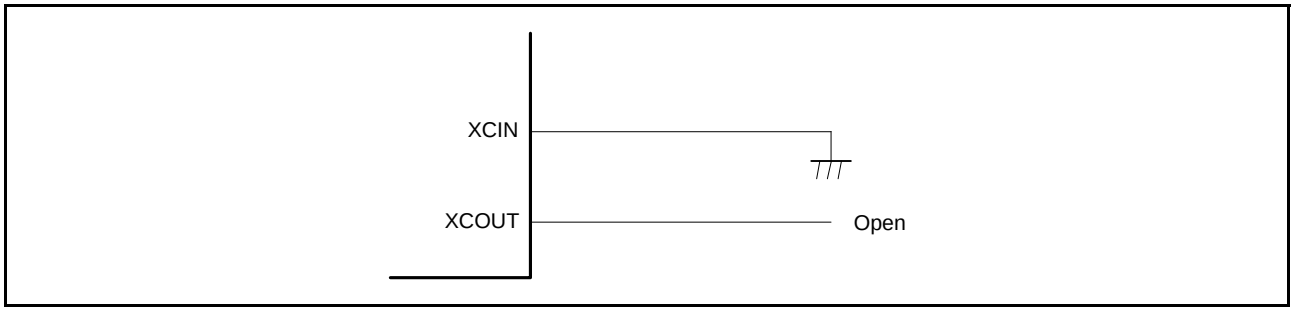


Figure 9.7 Pin Handling when Sub-Clock is not Used

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.
An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU output can be forcedly driven to the high-impedance on the detection. For details, see section 22, Multi-Function Timer Pulse Unit 2 (MTU2a) and section 23, Port Output Enable 2 (POE2a).
In RX63N/RX631 Group the main clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, due to a malfunction of the main clock oscillator (see Table 49.27,Oscillation Stop Detection 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, if an oscillation stop is detected with the main clock or PLL clock selected as the system clock source, 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 cleared 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 cleared 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 a specified oscillation settling time has elapsed.
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, and CAN clock (CANCLK), which are provided as the system clock sources. The main clock as the RTC-dedicated main clock source (RTCMCLK) is not switched to the low-speed clock.
The system clock (ICLK) frequency during the low-speed clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock select bits (SCKCR.ICK[3:0])

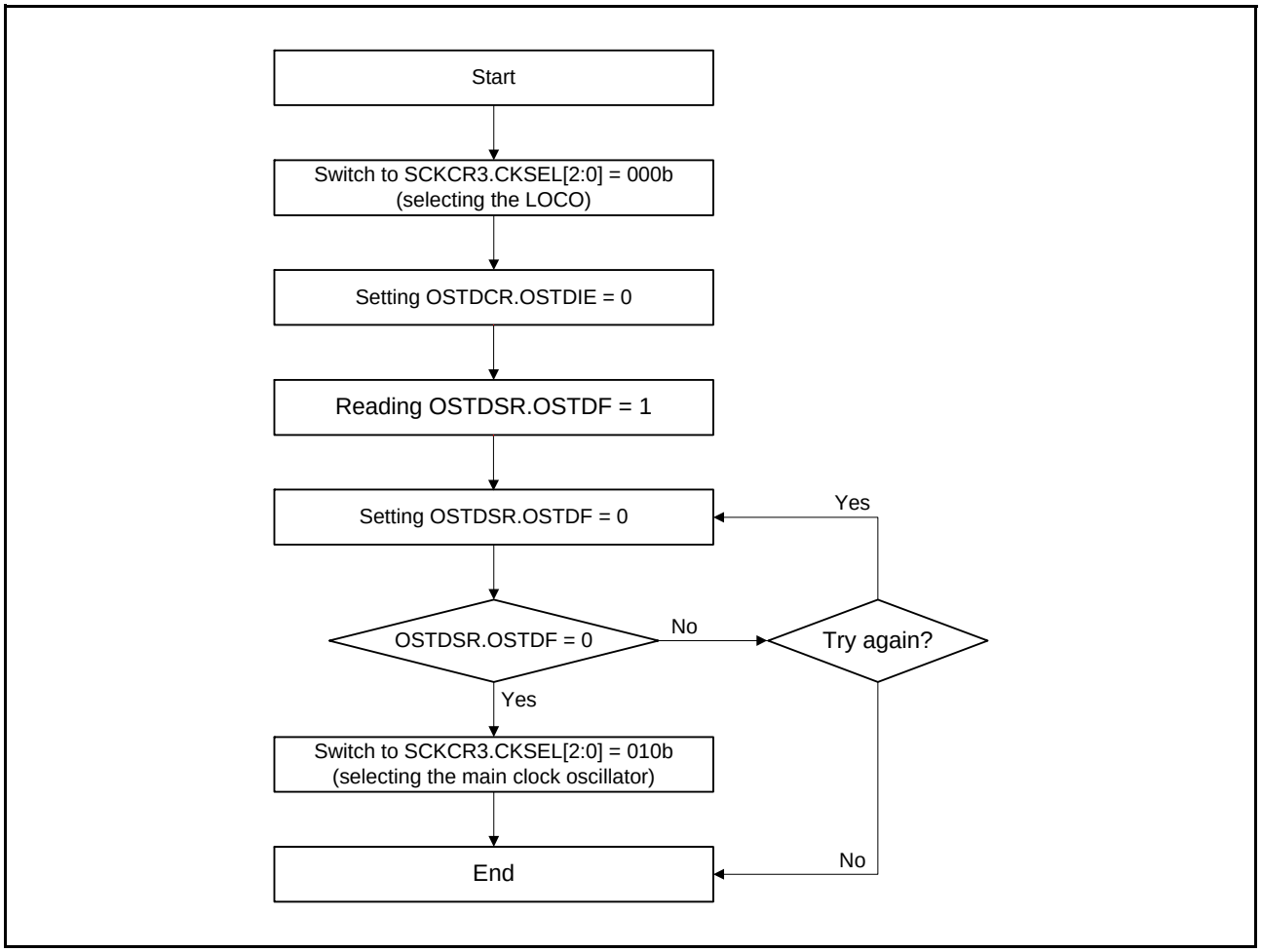


Figure 9.8 Flow of Recovery from Detection of Oscillator Stop

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 2 (POE). On accepting the notification of the oscillation stop, the POE sets the OSTST high-impedance flag in input level control/status register 3 (ICSR3.OSTSTF) to 1. After the oscillation stop is detected, wait for at least 10 cycles of PCLK before writing to this ICSR3.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.

The oscillation stop detection interrupt is a non-maskable interrupt. Since non-maskable interrupts are disabled in the initial state after a reset release, enable the non-maskable interrupts by the software before using oscillation stop detection interrupts. For details, see section 15, Interrupt Controller (ICUb).

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 low-speed clock for the IWDT, and the external clock for JTAG. The 13 internal clocks listed below are produced from these sources.

(1) Operating clock of the CPU, DMAC, DTC, ROM, and RAM: System clock (ICLK)
(2) Operating clock of ETHERC and EDMAC modules: Peripheral module clock (PCLKA)
(3) Operating clock of peripheral modules other than ETHERC and EDMAC: Peripheral module clock (PCLKB)
(4) Operating clock of the FlashIF: FlashIF clock (FCLK)
(5) Clock for the external bus controller and external pin output: External bus clock (BCLK)
(6) Clock for the external bus controller and SDRAM external pin output: SDRAM clock (SDCLK)
(7) Operating clock for the USB: USB clock (UCLK)
(8) Operating clock for the CAN: CAN clock (CANCLK)
(9) Operating clock for the IEBUS: IEBUS clock (IECLK)
(10) Operating clock for the RTC: RTC-dedicated sub-clock (RTCSCLK)
(11) Operating clock for the IWDT: IWDT-dedicated low-speed clock (IWDTCLK)
(12) Operating clock for the JTAG: JTAG clock (JTAGTCK)

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], and PCKB[3:0] bits in SCKCR, the UCK[3:0] and IEBCK[3:0] bits in SCKCR2, the clock source selected by the CKSEL[2:0] bits in SCKCR3, and the bits that select the frequency of the PLL circuit (STC[5:0] and PLIDIV[1:0] in PLLCR). If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

9.7.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DMAC, DTC, ROM, and RAM.
The ICLK frequency is specified by the ICK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

9.7.2 Peripheral Module Clock

The peripheral module clocks (PCLKA) are the operating clocks for use by peripheral modules.
One of the peripheral module clock (PCLKA) is for use by the ETHERC and EDMAC modules. The other peripheral module clock (PCLKB) is for use by the other modules.
The PCLKA frequencies are specified by the PCKA[3:0] and PCKB[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

9.7.3 FlashIF Clock

The FlashIF clock (FCLK) is used as the operating clock of the FlashIF.
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.

9.7.4 External Bus Clock

The external bus clock (BCLK) is an operating clock for the external bus controller. It is also output externally from the BCLK pin for the external connection 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.

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 cleared to 0 and the PFBCR1.SDCLKE bit is set to 1, 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 SDRAM clock frequency is specified by the BCK[3:0] bits in SCKCR.
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 an operating clock for the USB.
The UCLK frequency is specified by the UCK[3:0] bits in SCKCR2, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.
A 48-MHz clock must be supplied to the USB module. When the USB module is used, setting must be made so that UCLK is 48 MHz.

9.7.7 CAN Clock

The CAN clock (CANCLK) is an operating clock for the CAN.
CANCLK is generated by the main clock oscillator.

9.7.8 IEBUS Clock

The IEBUS clock (IECLK) is an operating clock for the IEBUS.
The IECLK frequency is specified by the IEBCK[3:0] bits in SCKCR2, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

9.7.9 RTC-Dedicated Clock

The RTC-dedicated 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.

9.7.10 IWDT-Dedicated Low-Speed Clock

The IWDT-dedicated low-speed clock (IWDTCLK) is the operating clock for the IWDT.
IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

9.7.11 JTAG Clock

The JTAG-dedicated clock (JTAGTCK) is the operating clock for the JTAG.
JTAGTCK is generated by the external clock for JTAG (TCK).

9.8 Pin Settings When an Oscillator is Connected

(1) Main clock

Set P36 and P37 as input pins and then clear the main clock oscillator stopping bit (MOSTP in MOSCCR) to 0 (so that the clock runs).

(2) Sub-clock

Clear the sub-clock oscillator stopping bit (SOSTP in SOSCCR) to 0 (so that the clock runs).

9.9 Pin Settings When an External Clock is Connected

(1) Main clock

Set P36 and P37 as input pins and then clear the main clock oscillator stopping bit (MOSTP in MOSCCR) to 0 (so that the clock runs).

9.10 Usage Notes

9.10.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clocks (PCLKA and PCLKB), FlashIF 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 PCLKA and PCLKB. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

(2) The following relation is required between the frequencies of the system clock (ICLK) and external bus clock (BCLK).
ICLK ≥ BCLK

(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) To secure the processing after the clock frequency is changed, modify the pertinent clock control register to change the frequency, and then read the value from the register, and then perform the subsequent processing.

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 9.9 to prevent electromagnetic induction from interfering with correct oscillation.

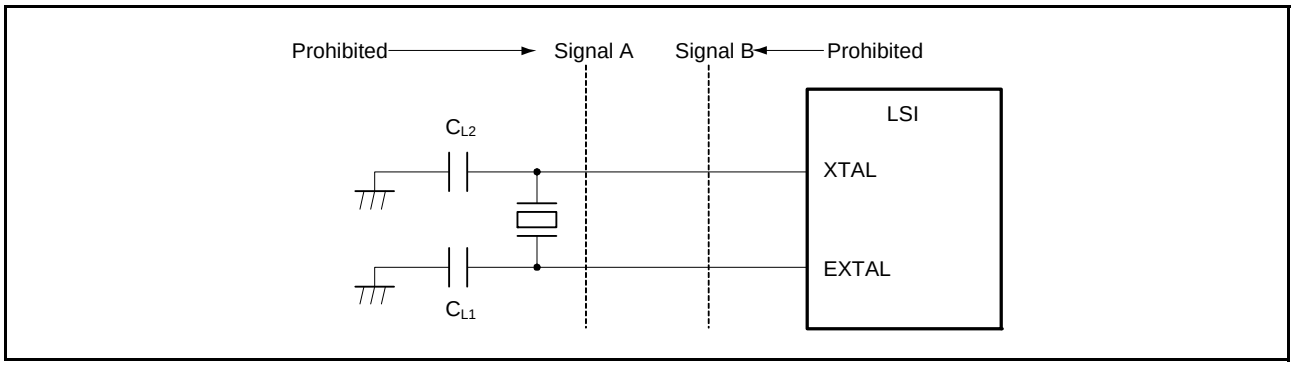


Figure 9.9 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)

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 20.36, 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 or USB boot mode*1

Note 1. For the conditions on the oscillator in the various modes, see item 9 in section 42.14, Usage Notes.

9.10.5 Notes on Sub-Clock

The sub-clock can be used as the system clock, as the source to drive counting by the real-time 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.

- With regard to making the sub-clock oscillator run or stop, setting either the sub-clock oscillator stop bit in the sub-clock oscillator control register (SOSCCR.SOSTP) or the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) will make the oscillator run.
- The SOSCCR.SOSTP bit should be used to set the sub-clock as the system clock; and the RCR3.RTCEN bit should be used to set the sub-clock as the realtime clock count source.
- Since the sub-clock control circuit is in an unstable state after a cold start, it need be initialized regardless of whether it is to be used or not. Initialize the sub-clock by setting both of the SOSCCR.SOSTP bit and the RCR3.RTCEN bit to the "stop" value. For details on initialization of the RCR3.RTCEN bit, see section 28, Realtime Clock (RTCa).
- If the RCR3.RTCEN bit is modified after the SOSCCR.SOFTP bit is modified, or if the SOSCCR.SOFTP 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.

10. Frequency Measurement Circuit (MCK)

The respective counter-clock extension circuits of the internal MTU (system 1) or unit 0 of TPU (system 2) can be used to monitor the main clock, sub-clock, LOCO (for system use), PLL, and HOCO for abnormal frequencies.

10.1 Overview

Table 10.1 lists the specifications of the frequency measurement circuit. Figure 10.1 and Figure 10.2 illustrate block diagrams (of system 1 and system 2), and Figure 10.3 illustrates the software configuration.

Table 10.1 Specifications of Frequency Measurement Circuit

Particulars			Description
Clock to be monitored			Main clock, sub clock, LOCO (system), PLL, HOCO*1
Circuit structure			• Channel 0 (MTU0) and channel 1 (MTU1) of MTU, counter-clock extension circuit 1 (system 1) • Channel 0 (TPU0) and channel 1 (TPU1) of TPU unit 0, counter-clock extension circuit 2 (system 2)
Settings	Counter-clock extension register n (n = 1, 2)		Select clock to be monitored from the main clock, sub clock, and LOCO.*2
	MTU0/ TPU0	Timer mode	Normal operation (compare match)
		Counter clock	MTCLKD/TCLKD external pin input (clock selected by counter-clock extension register n; n = 1, 2)
		Compare match	Compare match A, output disabled
		Counter clear	Cleared by compare match A
		Interrupt	Not used
	MTU1/ TPU1	Timer mode	Normal operation (input capture)
		Counter clock	PCLK
		Input capture	Input capture A by compare match A of channel 0
		Counter clear	Cleared by input capture A
		Interrupt	Input capture A (TGI1A)
Minimum pulse width for input signals on the MTCLKD/TCLKD external pin (output signals for counter-clock extension circuit n; n = 1, 2)			• Single edge specification: tTCKWH minimum cycle = 1.5 × (1/PCLK) tPcyc • Double edge specification: tTCKWL minimum cycle = 2.5 × (1/PCLK) tPcyc (tTCKWH, tTCKWL: Timer clock pulse width, tPcyc: PCLK cycle)
Power consumption reducing function			Module stop state can be set.
Others			System 1 and system 2 are capable of operating independently.

Note 1. PLL and HOCO can be selected as a clock to be monitored when they are system clocks (clock source of PCLK)
Note 2. When PLL is selected as a clock to be monitored, a clock other than the main clock should be selected by the counter-clock extension register n (n = 1, 2).

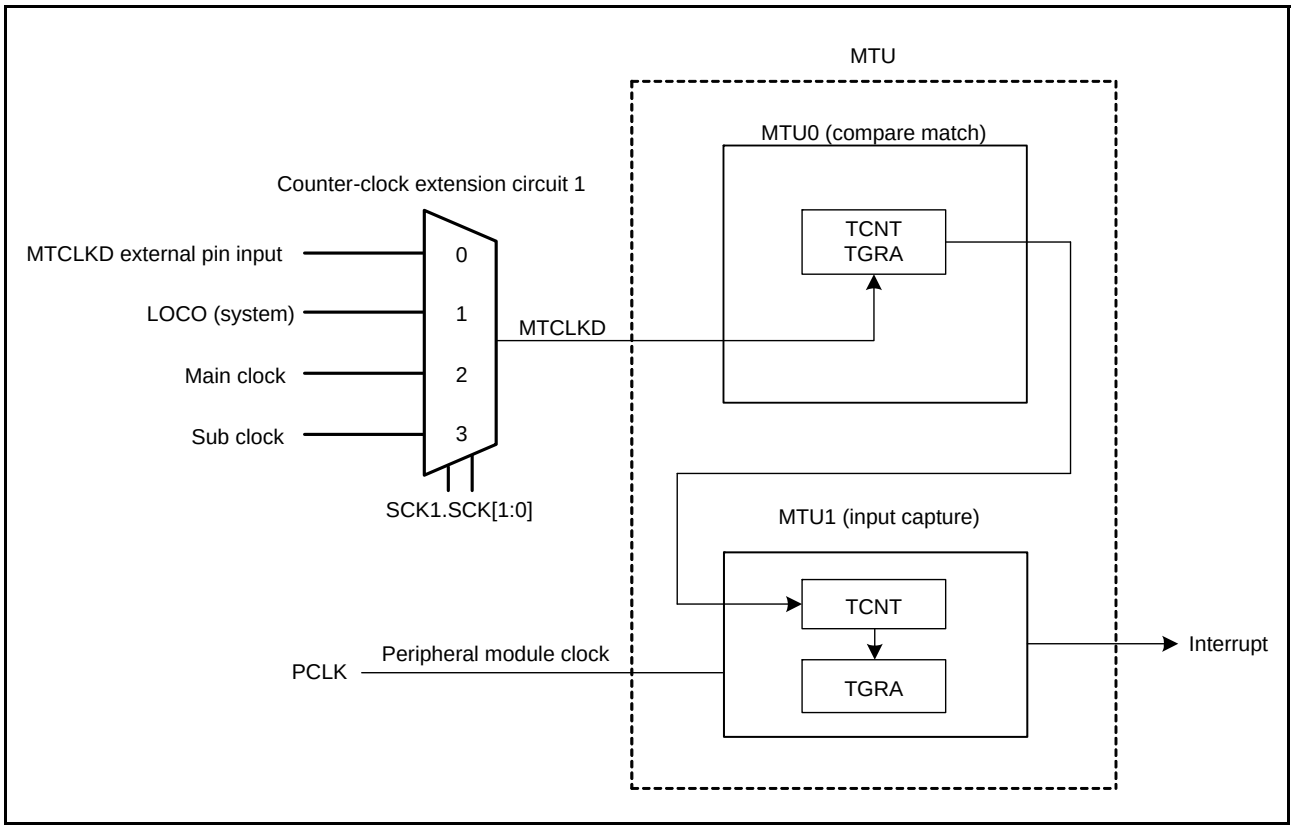


Figure 10.1 Block Diagram of Frequency Measurement Circuit with MTU (System 1)

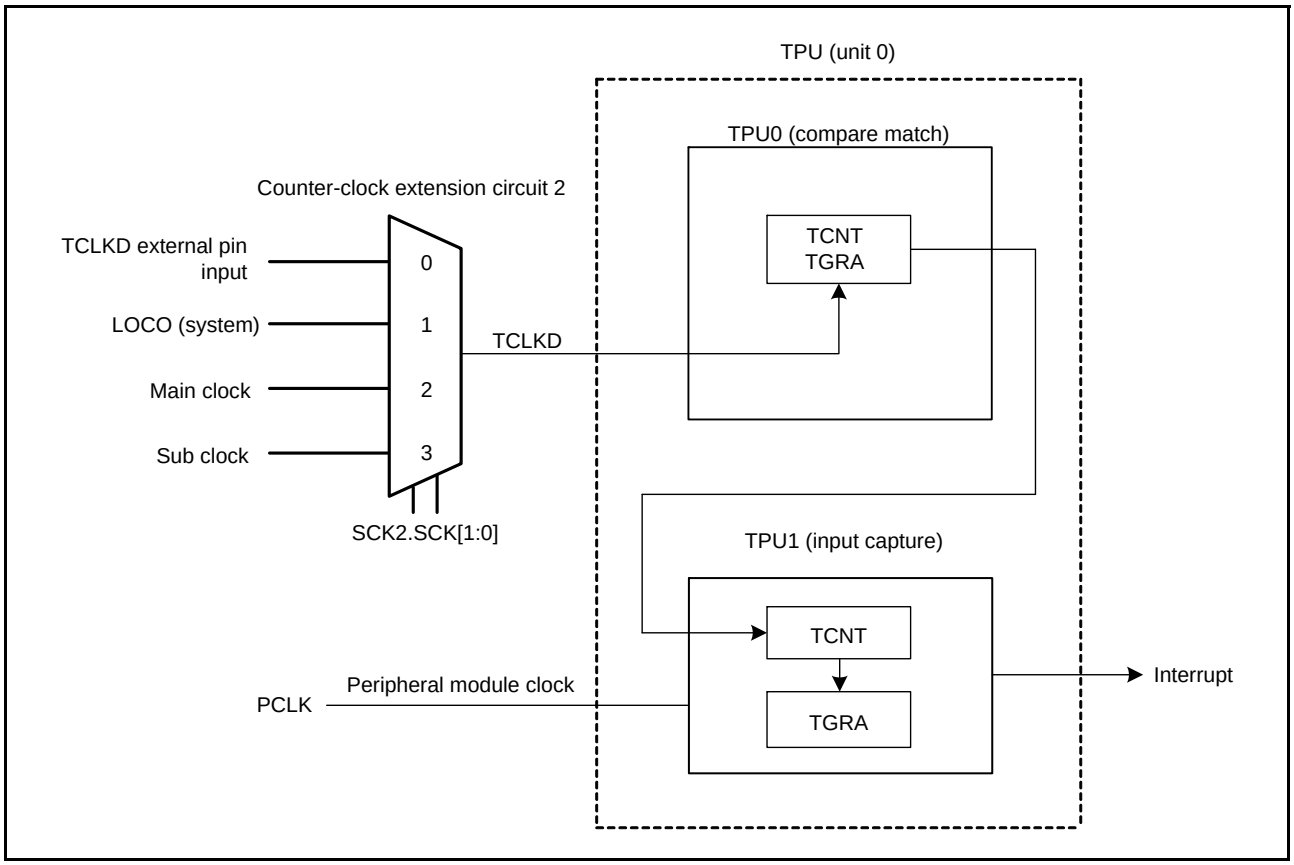
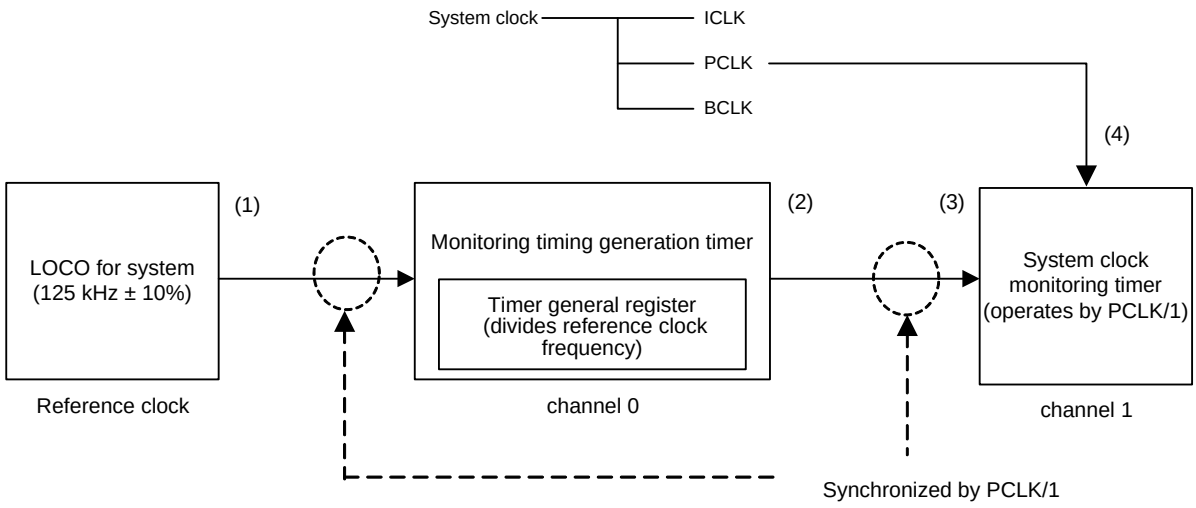


Figure 10.2 Block Diagram of Frequency Measurement Circuit with TPU Unit 0 (System 2)



(1) Reference clock is set as clock source of channel 0.
(2) Compare match output of channel 0 is used to divide the reference clock frequency.
(3) Frequency-divided reference clock (compare match output) is used as input capture timing of channel 1.
(4) Whether the number of PCLK/1 counts is within permissible range is determined every time input capture interrupt of channel 1 occurs.

- The reference clock source is specified in counter-clock extension register n (n = 1, 2).
- The input on the MTCLKD/TCLKD external pin is specified for the counter clock of channel 0.
- Number of channel 1 counts should be set so as to obtain a value of 1,000 counts or more.
- When HOCO is selected as the clock to be monitored, HOCO should be set as clock source of PCLK. In this case, any clock can be set as reference clock.

Figure 10.3 Software Configuration

10.2 Register Descriptions

10.2.1 Counter-Clock Extension Register n (n = 1, 2)

Address(es): 0008 C880h: SCK1, 0008 C890h: SCK2

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

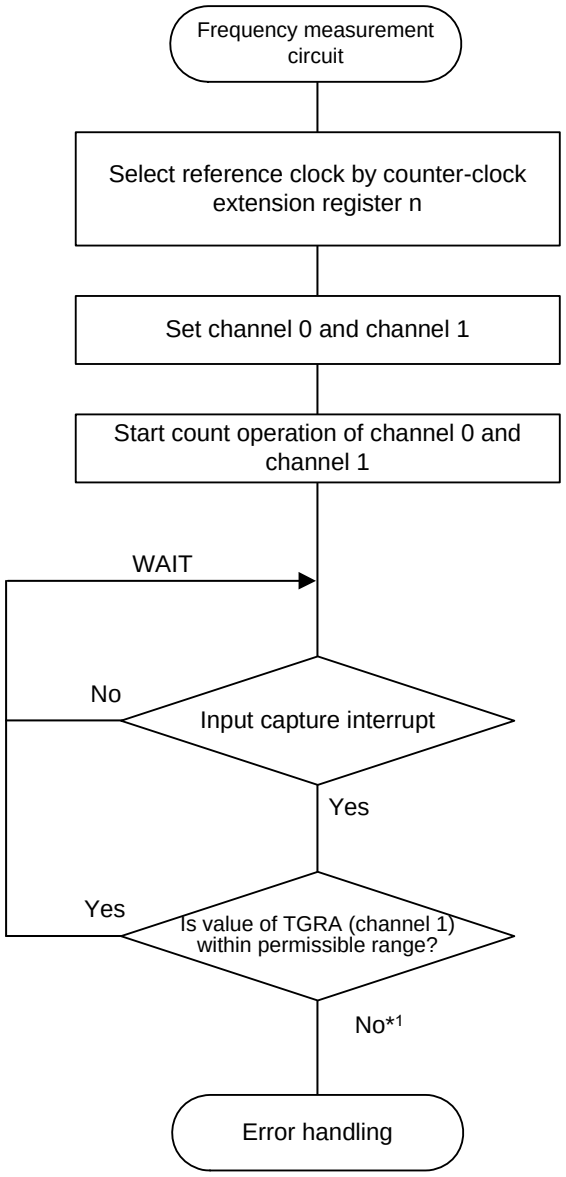
Bit	Symbol	Bit Name	Description	R/W
b1, b0	SCK[1:0]	Standard Clock Select	b1 b2 0 0: MTCLKD/TCLKD external pin input*1 0 1: LOCO 1 0: Main clock 1 1: Sub clock	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Select MTCLKD external pin input for the SCK1 register and TCLKD external pin input for the SCK2 register.

SCK1 is used to select the counter clock of channel 0 of MTU (MTU0).
SCK2 is used to select the counter clock of channel 0 of TPU unit (TPU0).
In combination with SCKn register setting, select the input on the MTCLKD/TCLKD external pin at the counter clock for the counter of channel 0. Selecting the input on the MTCLKD/TCLKD external pin causes the clock signal selected in the SCKn register to become the counter clock for MTU0.
If the MTCLKD or TCLKD external pin is selected as the source of the clock for counting by the counter of a channel other than channel 0, the actual counter clock for the given channel becomes the signal selected in the SCKn register.

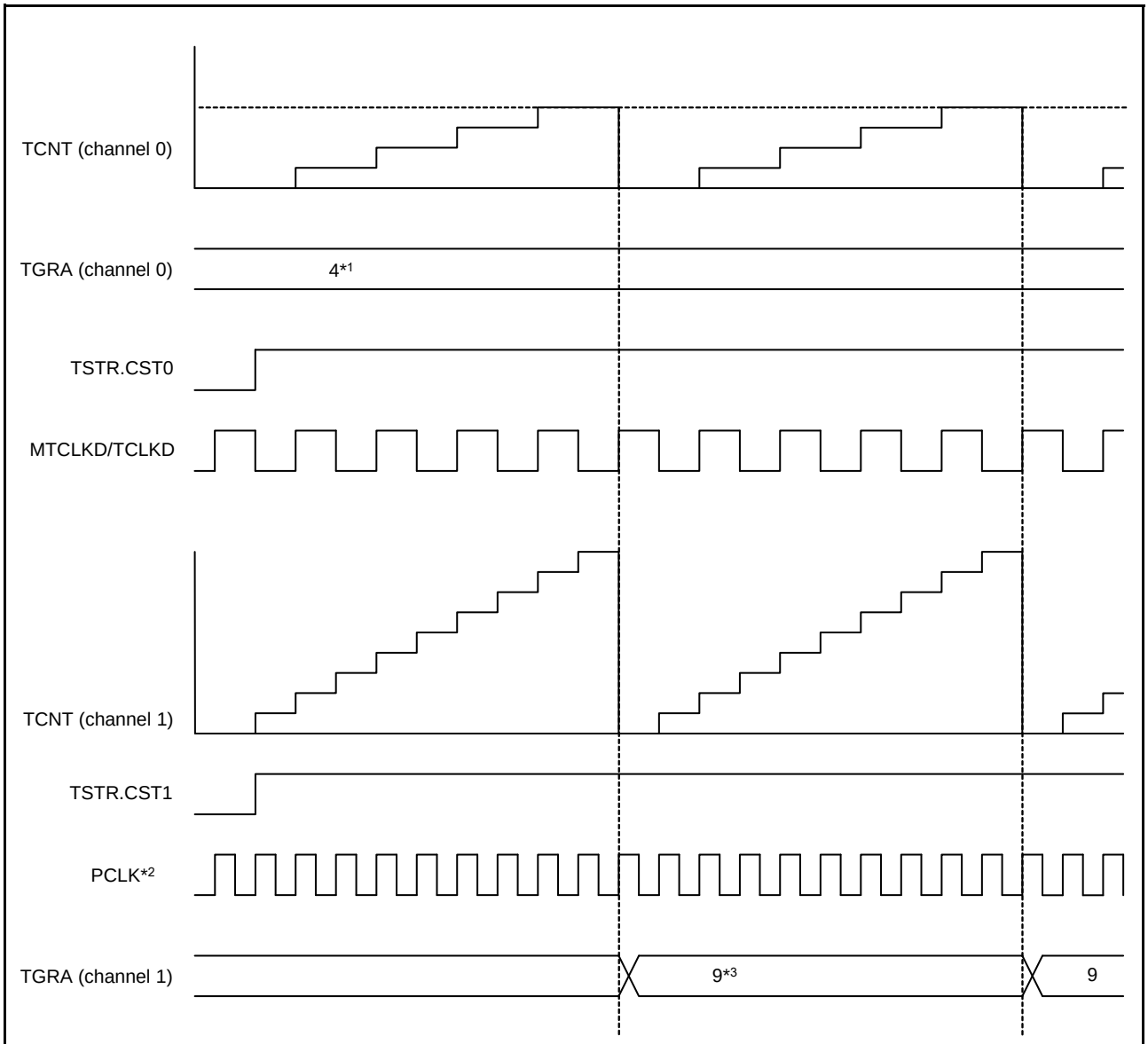
10.3 Operation

Figure 10.4 illustrates a flowchart of the frequency measurement circuit. Figure 10.5 illustrates an example of the MTU/TPU operation.



Note 1. The first TGRA (channel 1) value should not be used as it is undefined (variable).

Figure 10.4 Operation Flow of Frequency Measurement Circuit



TCNT: Timer counter
TGRA: Timer general register A
CSTi: Counter start i bit in the timer start register

Note 1. Set an appropriate value in TGRA (channel 0) for the intended use.
Note 2. Set an appropriate frequency division value of PCLK for the intended use.
Note 3. The first value captured in TGRA (channel 1) should not be used as it is undefined (variable).

Figure 10.5 Example of MTU/TPU Operation

10.4 Usage Notes

10.4.1 Setting the Module-Stop Control Registers

The module-stop control registers can be used to place modules in and release modules from the module-stopped state. The several modules that realize frequency measurement are all stopped in their initial state. Releasing the modules from the stopped state makes operations for frequency measurement possible. The method of release is as described below.
Clearing the module stop setting bit (MSTPA9) for multi-function timer pulse unit 2 (unit 0) in module stop control register A (MSTPCRA) makes the registers of MTU unit 0 and of counter-clock extension circuit 1 accessible.
Clearing the module stop setting bit (MSTPA13) for 16-bit timer pulse unit 0 (unit 0) in module stop control register A (MSTPCRA) makes the registers of TPU unit 0 and of counter-clock extension circuit 2 accessible.
Clearing the module stop setting bit (MSTPC19) for the frequency measurement circuit in module stop control register C (MSTPCRC) enables input of the reference clock signals for counter-clock extension circuits 1 and 2.
For details on the module-stop control registers, refer to section 11, Low Power Consumption.

11. Low Power Consumption

11.1 Overview

The RX63N/RX631 Group has several functions for reducing power consumption, including switching of clock signals to reduce power consumption, BCLK output control, SDCLK output control function, 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 canceling each mode.
After a reset, this LSI enters the normal program execution state, but modules except for the DMAC, DTC, and on-chip 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), 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. • 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 (1/2)

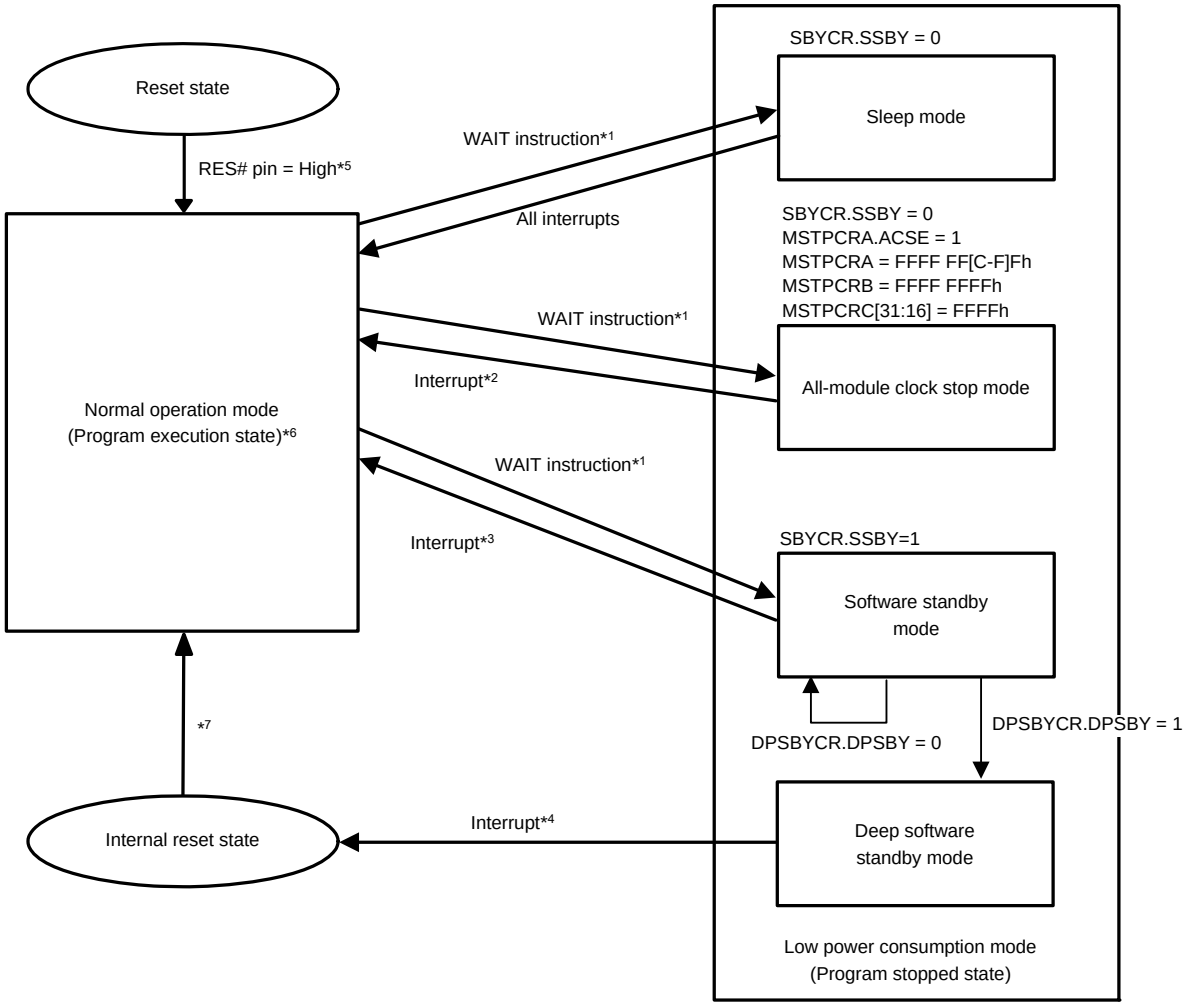
Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition condition	Control register + instruction	Control register + instruction	Control register + instruction	Control register + instruction
Canceling method other than reset	Interrupt	Interrupt*1	Interrupt*2	Interrupt*3
State after cancellation*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)
On-chip RAM1 (0001 0000h to 0001 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
On-chip RAM0 (0000 0000h to 0000 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained/Undefined)*8
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)

Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode (2/2)

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
USB 2.0 host/function module (USB)	Operating possible	Stopped*9	Stopped	Stopped (Retained/Undefined)*10
Watchdog timer (WDT)	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 (LVD)	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 interval, IWDT, USB suspend/resume, voltage-monitoring 1, voltage-monitoring 2, and oscillator-stopped detection interrupts).
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the RTC alarm, RTC interval, 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 interval, 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. A pin with a name having the suffix “-DS” can be used to initiate release from deep software standby mode.
Note 4. This does not include release initiated by a reset on the RES# pin, 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 forcible oscillation control bit (MOFXIN) in the main clock oscillator forcible oscillation control register (MOFCR).
Note 6. Operation or stopping is selected by the sub-clock 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 counter 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 (DEEPCUT[1:0]) in the deep standby control register (DPSBYCR).
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 (DEEPCUT[1:0]) in the deep standby control register (DPSBYCR). 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.
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 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#, and ALE) 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 interval, 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 interval, 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 interval, 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.

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 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR). Also in the case of recovery from sleep mode, the clock source will be the same before and at the time of the transition.

Note 7. When an interrupt request 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 a reset from a pin, 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 to cancel 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#, and ALE) 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 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 (OSTDCR.OSTDE) is 1, setting of the SSBY bit is invalid. 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.

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	—	MSTPA29	MSTPA28	MSTPA27	—	—	MSTPA24	MSTPA23	—	—	—	MSTPA19	—	MSTPA17	—
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
	MSTPA15	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9	—	—	—	MSTPA5	MSTPA4	—	—	—	—
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
b3 to b0	—	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: The module stop state is canceled 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: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b8 to b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 2 Module Stop	Target module: MTU (MTU0 to MTU5) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b10	MSTPA10	Programmable Pulse Generator (Unit 1) Module Stop	Target module: PPG1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b11	MSTPA11	Programmable Pulse Generator (Unit 0) Module Stop	Target module: PPG0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b12	MSTPA12	16-Bit Timer Pulse Unit 1 (Unit 1) Module Stop	Target module: TPU unit 1 (TPU6 to TPU11) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b13	MSTPA13	16-Bit Timer Pulse Unit 0 (Unit 0) Module Stop	Target module: TPU unit 0 (TPU0 to TPU5) 0: The module stop state is canceled 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: The module stop state is canceled 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: The module stop state is canceled 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	MSTPA17	12-bit A/D Converter Module Stop	Target module: S12AD 0: The module stop state is canceled 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	D/A Converter Module Stop	Target module: DA 0: The module stop state is canceled 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	MSTPA23	10-bit A/D Converter (Unit 0) Module Stop	Target module: AD0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W

Bit	Symbol	Bit Name	Description	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: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b29	MSTPA29	EXDMA Controller Module Stop	Target module: EXDMAC 0: The module stop state is canceled 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, and MSTPCRC 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 MSTPA 4 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.

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	—	MSTPB21	MSTPB20	MSTPB19	MSTPB18	MSTPB17	MSTPB16
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	—	—	—	—	—	—	MSTPB8	—	—	—	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: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b1	MSTPB1	CAN Module 1 Module Stop*1	Target module: CAN1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b2	MSTPB2	CAN Module 2 Module Stop*1	Target module: CAN2 0: The module stop state is canceled 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

Bit	Symbol	Bit Name	Description	R/W
b4	MSTPB4	Serial Communication Interface SCId Module Stop	Target module: SCId (SCI12) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b8	MSTPB8	Temperature Sensor Module Stop	Target module: Temperature sensor 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b14 to b9	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b15	MSTPB15	Ethernet Controller DMAC Module Stop	Target module: EDMAC 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b16	MSTPB16	Serial Peripheral Interface 1 Module Stop	Target module: RSPI1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b18	MSTPB18	Universal Serial Bus Interface (Port 1) Module Stop*2	Target module: USB1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b19	MSTPB19	Universal Serial Bus Interface (Port 0) Module Stop*2	Target module: USB0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b20	MSTPB20	I^2C Bus Interface 1 Module Stop	Target module: RIIC1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: The module stop state is canceled 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	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b24	MSTPB24	Serial Communication Interface 7 Module Stop	Target module: SCI7 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b27	MSTPB27	Serial Communication Interface 4 Module Stop	Target module: SCI4 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop	Target module: SCI3 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SCI0 0: The module stop state is canceled 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 MSTPBi is stabilized. For entering software

standby mode after rewriting the MSTPBi bit, wait for two CAN clock (CANCLK) cycles after rewriting, and execute the WAIT instruction (i = 0 to 2).

Note 2. For entering software standby mode after rewriting the MSTPB1i bit, wait for two USB clock (UCLK) cycles after rewriting, and execute the WAIT instruction.

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	—	MSTPC22	—	—	MSTPC19	MSTPC18	MSTPC17	MSTPC16
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC1	MSTPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

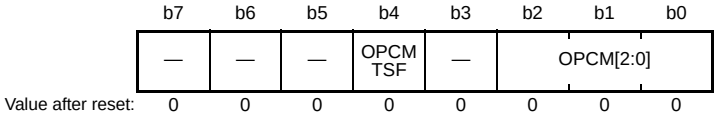
Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0	RAM0 Module Stop*1	Target module: RAM0 (0000 0000h to 0000 FFFFh) 0: RAM0 operating 1: RAM0 stopped	R/W
b1	MSTPC1	RAM1 Module Stop*1	Target module: RAM1 (0001 0000h to 0001 FFFFh) 0: RAM1 operating 1: RAM1 stopped	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	MSTPC16	I^2C Bus Interface 3 Module Stop	Target module: RIIC3 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b17	MSTPC17	I^2C Bus Interface 2 Module Stop	Target module: RIIC2 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b18	MSTPC18	IEBUS Module Stop*2	Target module: IEBUS 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b19	MSTPC19	Frequency Measurement Circuit Module Stop	Target module: MCK 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b21, b20	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b22	MSTPC22	Serial Peripheral Interface 2 Module Stop	Target module: RSPI2 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b23	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b24	MSTPC24	Serial Communication Interface 11 Module Stop	Target module: SCI11 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b25	MSTPC25	Serial Communication Interface 10 Module Stop	Target module: SCI10 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b26	MSTPC26	Serial Communication Interface 9 Module Stop	Target module: SCI9 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b27	MSTPC27	Serial Communication Interface 8 Module Stop	Target module: SCI8 0: The module stop state is canceled 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 MSTPC1 or MSTPC0 bit should not be set to 1 during access to the corresponding on-chip RAM. The corresponding on-chip RAM should not be accessed while the MSTPC1 or MSTPC0 bit is set to 1.

Note 2. For entering software standby mode after rewriting the MSTPC18 bit, wait for two IEBUS clock (IECLK) cycles after rewriting, and execute the WAIT instruction.

11.2.5 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OPCM[2:0]	Operating Power Control Mode Select	b2 b1 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

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 transition status flag (OPCMTSF) is 1 (operating power control mode switching is in progress)
- When the ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode) (i = 0 to 3)
- Period from the time of WAIT instruction issuance for a sleep mode transition, to return from sleep mode to normal operation

Writing to the flash memory while it is being programmed or erased is impossible because write access to the OPCCR register is not allowed.

For the procedure to use in shifting to operating power control mode, refer to section 11.5, Function for Lower Operating Power Consumption.

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
		FLASH Read					FLASH P/E	FLASH Read	FLASH P/E	
		ICLK	FCLK	PCLKA	PCLKB	BCLK	FCLK			
High-speed operating mode	000b	100 MHz max	50 MHz max	100MHz max	50 MHz max	100 MHz max	4 MHz to 50 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	1MHz max	1 MHz max	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 125 kHz	32 kHz to 125 kHz	125kHz max	125 kHz max	125 kHz max	P/E disabled	2.7 V to 3.6 V	P/E disabled	Low

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 100 MHz, and that of FCLK and PCLKB is 50 MHz. During FLASH programming/erasure (P/E), the FCLK can be operated in the operating frequency from 4 MHz to 50 MHz. The operating voltage is in the range of 2.7 to 3.6 V both for FLASH read and P/E. After a reset cancellation, 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 is disabled, and writing to set the PLLCR2.PLLWN 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 125 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 FLASHIF 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.

11.2.6 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 at sleep mode cancellation is disabled 1: Clock source switching at sleep mode cancellation is enabled	R/W

RSTCKCR is used to control clock source switching at cancellation of 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 clock oscillator control register (HOCOCR.HCSTP) corresponding to the clock source to be used on restoration are automatically modified to the operating state. However, the power of HOCO is not automatically switched on. 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]).
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 at sleep mode cancellation 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)

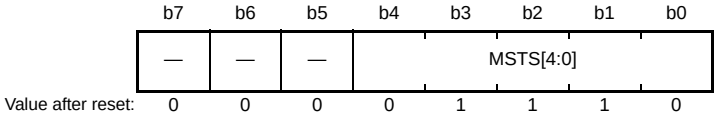
The RSTCKSEL[2:0] bits select the clock source to be used when sleep mode is canceled.
The clock source selected by the RSTCKSEL[2:0] bits is enabled only when the RSTCKEN bit is 1.

RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)

The RSTCKEN bit enables or disables clock source switching when sleep mode is canceled.
When sleep mode is canceled, 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.

11.2.7 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h

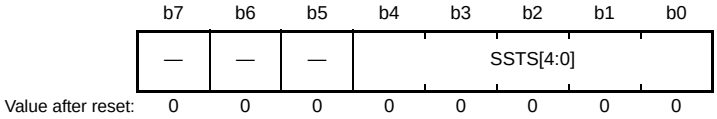


Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSTS[4:0]	Main Clock Oscillator Waiting Time	b4 b0 0 0 0 0 0: Waiting time = 2 cycles 0 0 0 0 1: Waiting time = 4 cycles 0 0 0 1 0: Waiting time = 8 cycles 0 0 0 1 1: Waiting time = 16 cycles 0 0 1 0 0: Waiting time = 32 cycles 0 0 1 0 1: Waiting time = 64 cycles 0 0 1 1 0: Waiting time = 512 cycles 0 0 1 1 1: Waiting time = 1024 cycles 0 1 0 0 0: Waiting time = 2048 cycles 0 1 0 0 1: Waiting time = 4096 cycles 0 1 0 1 0: Waiting time = 16384 cycles 0 1 0 1 1: Waiting time = 32768 cycles 0 1 1 0 0: Waiting time = 65536 cycles 0 1 1 0 1: Waiting time = 131072 cycles 0 1 1 1 0: Waiting time = 262144 cycles 0 1 1 1 1: Waiting time = 524288 cycles Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MOSCWTCR is used to control the oscillation settling time of the main clock oscillator.
Transmission of the main clock signal to the LSI starts after the number of cycles of the main clock set in the MOSCWTCR register have been counted.
Set the MSTS[4:0] bits so that the waiting time is at least as long as the main clock oscillator stabilization time (tMAINOSC). For example, if the frequency of the oscillator in use is 10 MHz (so that the period is 100 ns) and the MSTS[4:0] bits are set to 01101b, the waiting time will be 100 ns × 131072 cycles, which is approximately 13.12 ms.
The waiting time is not required when the main clock is externally input.
MOSCWTCR can only be rewritten when the MOSCCR.MOSTP bit is 1; do not rewrite MOSCWTCR with other settings.

11.2.8 Sub-Clock Oscillator Wait Control Register (SOSCWTCR)

Address(es): 0008 00A3h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	SSTS[4:0]	Sub-Clock Oscillator Waiting Time	b4 b0 0 0 0 0 0: Waiting time = 2s + 2 cycles 0 0 0 0 1: Waiting time = 2s + 4 cycles 0 0 0 1 0: Waiting time = 2s + 8 cycles 0 0 0 1 1: Waiting time = 2s + 16 cycles 0 0 1 0 0: Waiting time = 2s + 32 cycles 0 0 1 0 1: Waiting time = 2s + 64 cycles 0 0 1 1 0: Waiting time = 2s + 512 cycles 0 0 1 1 1: Waiting time = 2s + 1024 cycles 0 1 0 0 0: Waiting time = 2s + 2048 cycles 0 1 0 0 1: Waiting time = 2s + 4096 cycles 0 1 0 1 0: Waiting time = 2s + 16384 cycles 0 1 0 1 1: Waiting time = 2s + 32768 cycles 0 1 1 0 0: Waiting time = 2s + 65536 cycles 0 1 1 0 1: Waiting time = 2s + 131072 cycles 0 1 1 1 0: Waiting time = 2s + 262144 cycles 0 1 1 1 1: Waiting time = 2s + 524288 cycles Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SOSCWTCR is used to select the oscillation settling time of the sub-clock oscillator.

Set the SSTS[4:0] bits so that the waiting time is at least as long as the sub-clock oscillator stabilization time (tSUBOSC). For example, if the frequency of the oscillator in use is 32.768 kHz (so that the period is 30.5 μs) and the SSTS[4:0] bits are set to 01100b, the waiting time will be 2s + 30.5 μs × 65536 cycles, which is approximately 2s + 1s = 3s.

Writing to alter the setting in register SOSCWTCR is only possible when the SOSTP bit in SOSCCR is 1 (stopping the sub-clock oscillator). Do not attempt to change the setting if this is not the case.

11.2.9 PLL Wait Control Register (PLLWTCR)

Address(es): 0008 00A6h

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSTS[4:0]	PLL Waiting Time	b4 b0 0 0 0 0 0: Waiting time = 16 cycles 0 0 0 0 1: Waiting time = 32 cycles 0 0 0 1 0: Waiting time = 64 cycles 0 0 0 1 1: Waiting time = 512 cycles 0 0 1 0 0: Waiting time = 1024 cycles 0 0 1 0 1: Waiting time = 2048 cycles 0 0 1 1 0: Waiting time = 4096 cycles 0 0 1 1 1: Waiting time = 16384 cycles 0 1 0 0 0: Waiting time = 32768 cycles 0 1 0 0 1: Waiting time = 65536 cycles 0 1 0 1 0: Waiting time = 131072 cycles 0 1 0 1 1: Waiting time = 262144 cycles 0 1 1 0 0: Waiting time = 524288 cycles 0 1 1 0 1: Waiting time = 1048576 cycles 0 1 1 1 0: Waiting time = 2097152 cycles 0 1 1 1 1: Waiting time = 4194304 cycles Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PLLWTCR is used to select the oscillation settling time of the PLL.
Supply of the PLL clock signal within the LSI starts after the number of cycles of the PLL clock set in the PLLWTCR register have been counted.
A crystal oscillator is used as the oscillator. When PLL operation is selected without waiting for oscillation of the main clock to become stable, set the PSTS[4:0] bits so that the PLL oscillator stabilization time is at least as long as the PLL oscillator stabilization time (when the PLL is turned on without waiting for oscillation of the main clock to become stable, tPLL2). For example, if the frequency of the PLL oscillator is 200 MHz (period of 5 ns), the waiting time is 5 ns × 2097152 cycles ≈ 10.48 ms if the setting of the PSTS[4:0] bits is 01110b. When PLL operation is selected after waiting for oscillation of the main clock to become stable and the main clock is externally input, waiting for oscillation of the reference clock to become stable is not necessary. Therefore, set the PSTS[4:0] bits so that the waiting time (when the PLL is turned on after waiting for oscillation of the main clock to become stable, tPLL1) is at least as long as the PLL oscillator stabilization time.
PLLWTCR can only be rewritten when the PLLCR2.PLLEN bit is 1 (PLL stopped); do not rewrite PLLWTCR with other settings.

11.2.10 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 on-chip RAM (RAM0*1) and USB resume detecting unit in deep software standby mode 0 1: Power is not supplied to the on-chip RAM (RAM0*1) and USB resume detecting unit in deep software standby mode 1 0: Setting prohibited 1 1: Power is not supplied to the on-chip RAM (RAM0*1) 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: Deep software standby mode and I/O port retention are canceled simultaneously. 1: The I/O port state is retained even after deep software standby mode is canceled. 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

Note 1. For the on-chip RAM address space, see Table 11.2.

DPSBYCR is not initialized by the internal reset signal that is the source to cancel the 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 on-chip 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 on-chip RAM address space is divided into the RAM0 area and RAM1 area. For the on-chip RAM address space, see Table 11.2.

The internal power supply of RAM0 and USB resume detecting unit can be controlled by the setting of the DEEPCUT[1:0] bits.

When a USB suspend/resume interrupt is used as a deep software standby mode canceling source, the DEEPCUT[1:0] bits must be set to 00b.

When an LVD interrupt 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.

The internal power supply of RAM1 is stopped in deep software standby mode regardless of the setting of the DEEPCUT[1:0] bits.

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 deep software standby mode is canceled, or to cancel retaining 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 deep software standby mode is canceled by certain pins which are sources of external pin interrupts (NM1, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, and CRXI-DS) or a peripheral interrupt (RTC alarm, RTC interval, USB suspend/resume, voltage-monitoring 1, or voltage-monitoring 2). Write 0 to this bit to clear it.
The setting of the DPSBY bits 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 bits are 1, the transition after the execution of a WAIT instruction is to software standby mode.
The setting of the DPSBY bits 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 selected 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 bits are 1, the transition after the execution of a WAIT instruction is to software standby mode.

11.2.11 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: Canceling deep software standby mode by the IRQ0-DS pin is disabled 1: Canceling deep software standby mode by the IRQ0-DS pin is enabled	R/W
b1	DIRQ1E	IRQ1-DS Pin Enable	0: Canceling deep software standby mode by the IRQ1-DS pin is disabled 1: Canceling deep software standby mode by the IRQ1-DS pin is enabled	R/W
b2	DIRQ2E	IRQ2-DS Pin Enable	0: Canceling deep software standby mode by the IRQ2-DS pin is disabled 1: Canceling deep software standby mode by the IRQ2-DS pin is enabled	R/W
b3	DIRQ3E	IRQ3-DS Pin Enable	0: Canceling deep software standby mode by the IRQ3-DS pin is disabled 1: Canceling deep software standby mode by the IRQ3-DS pin is enabled	R/W
b4	DIRQ4E	IRQ4-DS Pin Enable	0: Canceling deep software standby mode by the IRQ4-DS pin is disabled 1: Canceling deep software standby mode by the IRQ4-DS pin is enabled	R/W
b5	DIRQ5E	IRQ5-DS Pin Enable	0: Canceling deep software standby mode by the IRQ5-DS pin is disabled 1: Canceling deep software standby mode by the IRQ5-DS pin is enabled	R/W
b6	DIRQ6E	IRQ6-DS Pin Enable	0: Canceling deep software standby mode by the IRQ6-DS pin is disabled 1: Canceling deep software standby mode by the IRQ6-DS pin is enabled	R/W
b7	DIRQ7E	IRQ7-DS Pin Enable	0: Canceling deep software standby mode by the IRQ7-DS pin is disabled 1: Canceling deep software standby mode by the IRQ7-DS pin is enabled	R/W

DPSIER0 is not initialized by the internal reset signal used as deep software standby mode canceling 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 cleared to 0 before a transition to deep software standby mode.

Even when DPSIER0 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR0 being set to 1. However, when DPSIEGR0 is 0, the rising edge is not detected, resulting in DPSIFR0 not being set to 1.

11.2.12 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: Canceling deep software standby mode by the IRQ8-DS pin is disabled 1: Canceling deep software standby mode by the IRQ8-DS pin is enabled	R/W
b1	DIRQ9E	IRQ9-DS Pin Enable	0: Canceling deep software standby mode by the IRQ9-DS pin is disabled 1: Canceling deep software standby mode by the IRQ9-DS pin is enabled	R/W
b2	DIRQ10E	IRQ10-DS Pin Enable	0: Canceling deep software standby mode by the IRQ10-DS pin is disabled 1: Canceling deep software standby mode by the IRQ10-DS pin is enabled	R/W
b3	DIRQ11E	IRQ11-DS Pin Enable	0: Canceling deep software standby mode by the IRQ11-DS pin is disabled 1: Canceling deep software standby mode by the IRQ11-DS pin is enabled	R/W
b4	DIRQ12E	IRQ12-DS Pin Enable	0: Canceling deep software standby mode by the IRQ12-DS pin is disabled 1: Canceling deep software standby mode by the IRQ12-DS pin is enabled	R/W
b5	DIRQ13E	IRQ13-DS Pin Enable	0: Canceling deep software standby mode by the IRQ13-DS pin is disabled 1: Canceling deep software standby mode by the IRQ13-DS pin is enabled	R/W
b6	DIRQ14E	IRQ14-DS Pin Enable	0: Canceling deep software standby mode by the IRQ14-DS pin is disabled 1: Canceling deep software standby mode by the IRQ14-DS pin is enabled	R/W
b7	DIRQ15E	IRQ15-DS Pin Enable	0: Canceling deep software standby mode by the IRQ15-DS pin is disabled 1: Canceling deep software standby mode by the IRQ15-DS pin is enabled	R/W

DPSIER1 is not initialized by the internal reset signal used as deep software standby mode canceling 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 cleared to 0 before a transition to deep software standby mode.

Even when DPSIER1 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR1 being set to 1. However, when DPSIEGR1 is 0, the rising edge is not detected, resulting in DPSIFR1 not being set to 1.

11.2.13 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 Cancel Signal Enable	0: Disable canceling deep software standby mode by the voltage-monitoring 1 signal 1: Enable canceling deep software standby mode by the voltage-monitoring 1 signal	R/W
b1	DLVD2IE	LVD2 Deep Standby Cancel Signal Enable	0: Disable canceling deep software standby mode by the voltage-monitoring 2 signal 1: Enable canceling deep software standby mode by the voltage-monitoring 2 signal	R/W
b2	DRTCIIE	RTC Interval Interrupt Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the RTC interval interrupt signal is disabled 1: Canceling deep software standby mode by the RTC interval interrupt signal is enabled	R/W
b3	DRTCAIE	RTC Alarm Interrupt Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the RTC alarm interrupt signal is disabled 1: Canceling deep software standby mode by the RTC alarm interrupt signal is enabled	R/W
b4	DNMIE	NMI Pin Enable	0: Canceling deep software standby mode by the NMI pin is disabled 1: Canceling deep software standby mode by the NMI pin is enabled	R/W*1
b5	DRIICDIE	SDA2-DS Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the SDA2-DS signal is disabled 1: Canceling deep software standby mode by the SDA2-DS signal is enabled	R/W
b6	DRIICCIE	SCL2-DS Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the SCL2-DS signal is disabled 1: Canceling deep software standby mode by the SCL2-DS signal is enabled	R/W
b7	DUSBIE	USB Suspend/Resume Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the USB suspend/resume is disabled 1: Canceling 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.

DPSIER2 is not initialized by the internal reset signal used as deep software standby mode canceling 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 cleared to 0 before a transition to deep software standby mode.

Even when DPSIER2 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR2 being set to 1. However, when DPSIEGR2 is 0, the rising edge is not detected, resulting in DPSIFR2 not being set to 1.

11.2.14 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 Cancel Signal Enable	0: Canceling deep software standby mode by the CRX1-DS pin is disabled 1: Canceling 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

DPSIER3 is not initialized by the internal reset signal used as deep software standby mode canceling 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 cleared to 0 before a transition to deep software standby mode.

Even when DPSIER3 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR3 being set to 1. However, when DPSIEGR3 is 0, the rising edge is not detected, resulting in DPSIFR3 not being set to 1 because the rising edge is not detected.

11.2.15 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 Deep Standby Cancel Flag	0: No cancel request by the IRQ0-DS pin is generated 1: A cancel request by the IRQ0-DS pin is generated	R(/W) *1
b1	DIRQ1F	IRQ1-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ1-DS pin is generated 1: A cancel request by the IRQ1-DS pin is generated	R(/W) *1
b2	DIRQ2F	IRQ2-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ2-DS pin is generated 1: A cancel request by the IRQ2-DS pin is generated	R(/W) *1
b3	DIRQ3F	IRQ3-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ3-DS pin is generated 1: A cancel request by the IRQ3-DS pin is generated	R(/W) *1
b4	DIRQ4F	IRQ4-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ4-DS pin is generated 1: A cancel request by the IRQ4-DS pin is generated	R(/W) *1
b5	DIRQ5F	IRQ5-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ5-DS pin is generated 1: A cancel request by the IRQ5-DS pin is generated	R(/W) *1
b6	DIRQ6F	IRQ6-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ6-DS pin is generated 1: A cancel request by the IRQ6-DS pin is generated	R(/W) *1
b7	DIRQ7F	IRQ7-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ7-DS pin is generated 1: A cancel request by the IRQ7-DS pin is generated	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR0 is generated.
Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER0 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR0 is cleared to 00h.
To clear DPSIFR0 to 00h after modifying DPSIER0, wait for at least six PCLKA, PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. Six or more PCLKA, PCLKB cycles can be secured, for example, by reading DPSIER0.
DPSIFR0 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DIRQnF Flags (IRQn Deep Standby Cancel Flag) (n = 0 to 7)

These flags indicate that a cancel request by the IRQn-DS pin has been generated.
[Setting condition]

- A cancel request by the IRQn-DS pin specified by DPSIEGR0 is generated

[Clearing condition]

- Each bit is read as 1 and then written by 0

11.2.16 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 Deep Standby Cancel Flag	0: No cancel request by the IRQ8-DS pin is generated 1: A cancel request by the IRQ8-DS pin is generated	R(/W) *1
b1	DIRQ9F	IRQ9-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ9-DS pin is generated 1: A cancel request by the IRQ9-DS pin is generated	R(/W) *1
b2	DIRQ10F	IRQ10-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ10-DS pin is generated 1: A cancel request by the IRQ10-DS pin is generated	R(/W) *1
b3	DIRQ11F	IRQ11-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ11-DS pin is generated 1: A cancel request by the IRQ11-DS pin is generated	R(/W) *1
b4	DIRQ12F	IRQ12-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ12-DS pin is generated 1: A cancel request by the IRQ12-DS pin is generated	R(/W) *1
b5	DIRQ13F	IRQ13-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ13-DS pin is generated 1: A cancel request by the IRQ13-DS pin is generated	R(/W) *1
b6	DIRQ14F	IRQ14-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ14-DS pin is generated 1: A cancel request by the IRQ14-DS pin is generated	R(/W) *1
b7	DIRQ15F	IRQ15-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ15-DS pin is generated 1: A cancel request by the IRQ15-DS pin is generated	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR1 is generated.
Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER1 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR1 is cleared to 00h.
To clear DPSIFR1 to 00h after modifying DPSIER1, wait for at least six PCLKA, PCLKB cycles, read DPSIFR1, and then write 0 to DPSIFR1. Six or more PCLKA, PCLKB cycles can be secured, for example, by reading DPSIER1.
DPSIFR1 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DIRQnF Flags (IRQn Deep Standby Cancel Flag) (n = 8 to 15)

These flags indicate that a cancel request by the IRQn-DS pin has been generated.
[Setting condition]

- A cancel request by the IRQn-DS pin specified by DPSIEGR1 is generated

[Clearing condition]

- Each bit is read as 1 and then written by 0

11.2.17 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 Cancel Flag	0: No cancel request by the voltage monitor 1 signal is generated 1: A cancel request by the voltage monitor 1 signal is generated	R(/W) *1
b1	DLVD2IF	LVD2 Deep Standby Cancel Flag	0: No cancel request by the voltage monitor 2 signal is generated 1: A cancel request by the voltage monitor 2 signal is generated	R(/W) *1
b2	DRTCIIF	RTC Interval Interrupt Deep Standby Cancel Flag	0: No cancel request by the RTC interval interrupt signal is generated 1: A cancel request by the RTC interval interrupt signal is generated	R(/W) *1
b3	DRTCAIF	RTC Alarm Interrupt Deep Standby Cancel Flag	0: No cancel request by the RTC alarm interrupt signal is generated 1: A cancel request by the RTC alarm interrupt signal is generated	R(/W) *1
b4	DNMIF	NMI Deep Standby Cancel Flag	0: No cancel request by the NMI pin is generated 1: A cancel request by the NMI pin is generated	R(/W) *1
b5	DRIICDIF	SDA2-DS Deep Standby Cancel Flag	0: No cancel request by the SDA2-DS signal is generated 1: A cancel request by the SDA2-DS signal is generated	R(/W) *1
b6	DRIICCIF	SCL2-DS Deep Standby Cancel Flag	0: No cancel request by the SCL2-DS signal is generated 1: A cancel request by the SCL2-DS signal is generated	R(/W) *1
b7	DUSBIF	USB Suspend/Resume Deep Standby Cancel Flag	0: No cancel request by the USB suspend/resume is generated 1: A cancel request by the USB suspend/resume is generated	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR2 is generated.
Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER2 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR2 is cleared to 00h.
To clear DPSIFR2 to 00h after modifying DPSIER2, wait for at least six PCLKA, PCLKB cycles, read DPSIFR2, and then write 0 to DPSIFR2. Six or more PCLKA, PCLKB cycles can be secured, for example, by reading DPSIER2.
DPSIFR2 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DLVDmIF Flag (LVDm Deep Standby Cancel Flag) (m = 1 or 2)

This flag indicates that a cancel request by the voltage monitor m signal has been generated.
[Setting condition]

- A cancel request is generated by the voltage-monitoring m signal that is selected in DPSIEGR2

[Clearing condition]

- This bit is read as 1 and then written by 0

DRTCIIF Flag (RTC Interval Interrupt Deep Standby Cancel Flag)

This flag indicates that a cancel request by the RTC interval interrupt signal has been generated.
[Setting condition]

- A cancel request by the RTC interval interrupt signal is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DRTCAIF Flag (RTC Alarm Interrupt Deep Standby Cancel Flag)

This flag indicates that a cancel request by the RTC alarm interrupt signal has been generated.
[Setting condition]

- A cancel request 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 Cancel Flag)

This flag indicates that a cancel request by the NMI pin has been generated.
[Setting condition]

- A cancel request by the NMI pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DRIICDIF Flag (SDA2-DS Deep Standby Cancel Flag)

This flag indicates that a cancel request by the SDA2-DS signal has been generated.
[Setting condition]

- A cancel request by the SDA2-DS pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DRIICCIF Flag (SCL2-DS Deep Standby Cancel Flag)

This flag indicates that a cancel request by the SCL2-DS signal has been generated.
[Setting condition]

- A cancel request by the SCL2-DS pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DUSBIF Flag (USB Suspend/Resume Deep Standby Cancel Flag)

This flag indicates that a cancel request by the USB suspend/resume has been generated.
[Setting condition]

- A cancel request by the USB suspend/resume is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

11.2.18 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 Cancel Flag	0: No cancel request by the CRX1-DS pin is generated 1: A cancel request by the CRX1-DS pin is generated	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 cancel request specified by DPSIEGR3 is generated.
Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER3 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR3 is cleared to 00h.
To clear DPSIFR3 to 00h after modifying DPSIER3, wait for at least six PCLKA, PCLKB cycles, read DPSIFR3, and then write 0 to DPSIFR3. Six or more PCLKA, PCLKB cycles can be secured, for example, by reading DPSIER3.
DPSIFR3 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DCANIF Flag (CAN Deep Standby Cancel Flag)

This flag indicates that a cancel request by the CAN-RXD pin has been generated.
[Setting condition]

- A cancel request by the CRX1-DS pin specified by DPSIEGR3 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

11.2.19 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 Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b1	DIRQ1EG	IRQ1-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b2	DIRQ2EG	IRQ2-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b3	DIRQ3EG	IRQ3-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b4	DIRQ4EG	IRQ4-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b5	DIRQ5EG	IRQ5-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6	DIRQ6EG	IRQ6-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b7	DIRQ7EG	IRQ7-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

DPSIEGR0 is not initialized by the internal reset signal that is the source to cancel the 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 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 Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b1	DIRQ9EG	IRQ9-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b2	DIRQ10EG	IRQ10-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b3	DIRQ11EG	IRQ11-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b4	DIRQ12EG	IRQ12-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b5	DIRQ13EG	IRQ13-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6	DIRQ14EG	IRQ14-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b7	DIRQ15EG	IRQ15-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

DPSIEGR1 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.21 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 Cancel request is generated when VCC < Vdet1 (fall) is detected 1: A Cancel request is generated when VCC ≥ Vdet1 (rise) is detected	R/W
b1	DLVD2EG	LVD2 Edge Select	0: A Cancel request is generated when VCC < Vdet2 (fall) is detected 1: A Cancel request 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 cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b5	DRIICDEG	SDA2-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6	DRIICCEG	SCL2-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

DPSIEGR2 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.22 Deep Standby Interrupt Edge Register 3 (DPSIEGR3)

Address(es): 0008 C28Dh

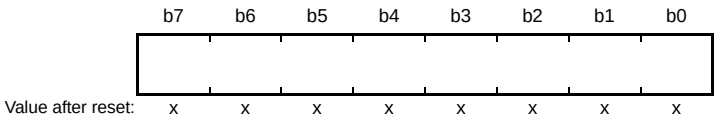
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DCANIEG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCANIEG	CRX1-DS Edge Select	0: A cancellation request is generated at a falling edge. 1: A cancellation request 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

DPSIEGR3 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.23 Deep Standby Backup Register (DPSBKRy) (y = 0 to 31)

Address(es): 0008 C2A0h to 0008 C2BFh



x: Undefined

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 on-chip 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], and PCKB[3:0] bits are set, the clock frequency changes. The CPU, DMAC, DTC, ROM, 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] bit.
The flash memory interface 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 C, i = 31 to 0) in MSTPCRA to MSTPCRC is set to 1, the specified module stops operating and enters the module stop state, but the CPU continues to operate independently. Clearing the MSTPmi bit to 0 cancels the module stop state, allowing the module to restart operating at the end of the bus cycle. The internal states of modules are retained in the module stop state.
After a reset is canceled, all modules other than the DMAC, DTC, and on-chip 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 operation mode to low-speed operation 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 OPCCR
↓
(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 operation mode 2 to high-speed operation mode

(Low-speed operation in the operating power consumption control mode used before mode-switching)
↓
Write to OPCCR
↓
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. 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 destination to be used for recovery from sleep mode to the CPU.
(3) Set the priority*2 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*2 for the 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, Interrupt Controller (ICUb).

11.6.1.2 Canceling Sleep Mode

Sleep mode is canceled by any interrupt, the reset signal from the RES# pin, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Canceling by an interrupt
  When an interrupt occurs, sleep mode is canceled and the interrupt exception handling 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), sleep mode is not canceled.
- Canceling by 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 handling.
- Canceling by a power-on reset
  Sleep mode is canceled by a power-on reset.
- Canceling by a voltage monitoring reset
  Sleep mode is canceled by a voltage monitoring reset from the voltage detection circuit.
- Canceling by the independent watchdog timer
  Sleep mode is canceled 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 sleep mode cannot be canceled by the independent watchdog timer reset.

Note 1. For details, see section 15, Interrupt Controller (ICUb).
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 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR). In addition, for details on the oscillation settling time, see section 11.2.7, Main Clock Oscillator Wait Control Register (MOSCWTCR), section 11.2.8, Sub-Clock Oscillator Wait Control Register (SOSCWTCR), and section 11.2.9, PLL Wait Control Register (PLLWTCR).

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, and MSTPCRC registers in the module stop state (MSTPCRA = FFFF FF[C-F]Fh*1, MSTPCRB = FFFF FFFFh, MSTPCRC[31:16] = 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*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. 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 destination to be used for recovery from all-module clock stop mode to the CPU.
(3) Set the priority*5 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*5 for the interrupt to be used for recovery from all-module clock stop mode to 1.
(5) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*4 of the CPU to 1).

Note 1. When the MSTPA15 or MSTPA14 bit in MSTPCRA is set, all-module clock stop mode can be entered even while the module is in operation state. However, the transition to all-module clock stop mode should be made in stopped state.
Note 2. The MSTPCRA.MSTPA4 and MSTPA5 bits select operation or stop of these modules.
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, Interrupt Controller (ICUb).

11.6.2.2 Canceling 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 interval, 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 activate the DTC or DMAC, the interrupt will not cancel all-module clock stop mode.

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 (ICUb).
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 cleared to 0, a transition to software standby mode is made.*1 In this mode, the CPU, on-chip peripheral functions, and all the oscillator functions stop. However, the contents of the CPU internal registers, on-chip 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.
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. 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, issue 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) Clear the PSW.I bit*2 of the CPU to 0.
(2) Set the interrupt destination 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*2 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*2 of the CPU to 1).

Note 1. With the product with 1.5-Mbyte ROM or more or the product with 176 pins or more, software standby mode is not entered even when a WAIT instruction is executed while the main clock oscillator or PLL circuit is selected as the clock source for the system clock. To enter software standby mode, select the low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), or sub-clock oscillator as the clock source before issuing the WAIT instruction.
Note 2. For details, see section 2, CPU.
Note 3. For details, see section 15, Interrupt Controller (ICUb).

11.6.3.2 Canceling 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 interval, 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 a software standby mode canceling source is generated, the oscillators which were operating before a transition to software standby mode restart operation. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

(1) Release due to an interrupt
When an interrupt request from the NMI, IRQ0 to IRQ15, RTC alarm, RTC cycle, 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 set by the MOSCWTCR.MSTS[4:0], SOSCWTCR.SSTS[4:0], or PLLWTCR.PSTS[4:0] bits has elapsed, the clock signal being supplied to the LSI is stable, so it is released from software standby mode and starts interrupt exception processing.

(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 11.2 shows an example where a transition to software standby mode is made at the falling edge of the IRQn pin, and software standby mode is canceled 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, software standby mode is canceled at the rising edge of the IRQn pin.
To return from software standby mode, settings of the interrupt controller (ICUb) are also necessary. For details, see section 15, Interrupt Controller (ICUb).

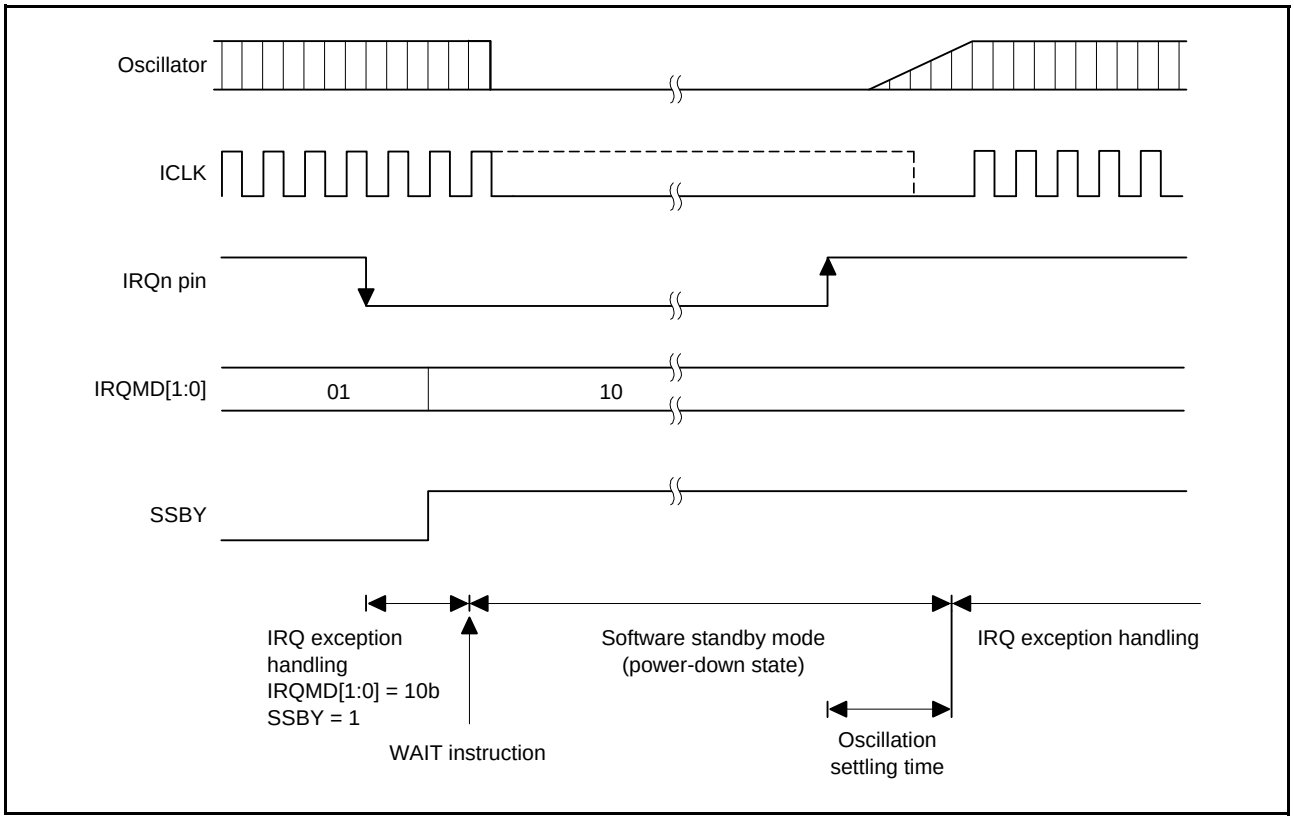


Figure 11.2 Example of Software Standby Mode Application

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 interval, SCL2-DS, SDA2-DS, CRX1-DS, and USB suspend/resume detecting unit), on-chip RAM1*2, and all functions of the oscillators are stopped; furthermore, since the internal supply of power for these modules is stopped, power consumption is markedly reduced. At this time, the contents of all the registers of the CPU and internal peripheral modules (except for parts of the RTC alarm, RTC interval, SCL2-DS, SDA2-DS, CRX1-DS, and USB suspend/resume detecting unit) become undefined. All data in the on-chip RAM1*2 become undefined, regardless of the setting of the DPSBYCR.DEEPCUT[1:0] bits.

Data in the on-chip RAM0*2 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 on-chip RAM0*2 and the section for detecting USB resumption is cut off, reducing power consumption. Data in the on-chip RAM0*2 become undefined at this time. If the setting of the DEEPCUT[1:0] bits is 11b, the internal supply of power to the on-chip RAM0*2, 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 49, 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 LOCO 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. In the same way, power supply to the IWDT dedicated LOCO 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.

Note 2. The on-chip RAM address space is divided into the RAM0 area and RAM1 area. For the on-chip RAM address space, see Table 11.2.

11.6.4.2 Canceling 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 interval, USB suspend/resume, voltage-monitoring 1, and voltage-monitoring 2 interrupts), an RES# pin reset, a power-on reset, or a voltage-monitoring 0 reset.

(1) Canceling by an external interrupt pin or internal interrupt signal
Cancellation of deep software standby mode is controlled by DPSIERn (n = 0 to 3) and DPSIFRn (n = 0 to 3). When a deep software standby canceling interrupt is generated, the corresponding flag in DPSIFRn is set to 1. At this time, if the canceling source is enabled in DPSIERn, deep software standby mode is canceled. Rising edge or falling edge can be selected by DPSIEGRn (n = 0 to 3). 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 canceling source is generated, the internal power supply and LOCO clock oscillation begin, and then the internal reset (deep software standby reset) is generated for the entire LSI.
A stable LOCO clock is then supplied to the entire LSI and deep software standby reset is canceled. At the same time, deep software standby mode is canceled and the reset exception handling starts.
When deep software standby mode is canceled 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
Deep software standby mode is canceled when the low level is applied to the RES# pin.
At this time, the RES# pin should be held low according to the specifications described in section 49, 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 when Deep Software Standby Mode is Canceled

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 when deep software standby mode is canceled. Upon cancellation of deep software standby mode, the reset exception handling starts immediately. 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 when deep software standby mode is canceled.
- When the DPSBYCR.IOKEEP bit= 1
  Although the inside of the LSI is initialized by an internal reset generated when deep software standby mode is canceled, 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 when deep software standby mode is canceled.

11.6.4.4 Example of Deep Software Standby Mode Application

Figure 11.3 shows an example where a transition to deep software standby mode is made at the falling edge of the IRQn-DS pin, and deep software standby mode is canceled 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, deep software standby mode is canceled at the rising edge of the IRQ-DS pin.

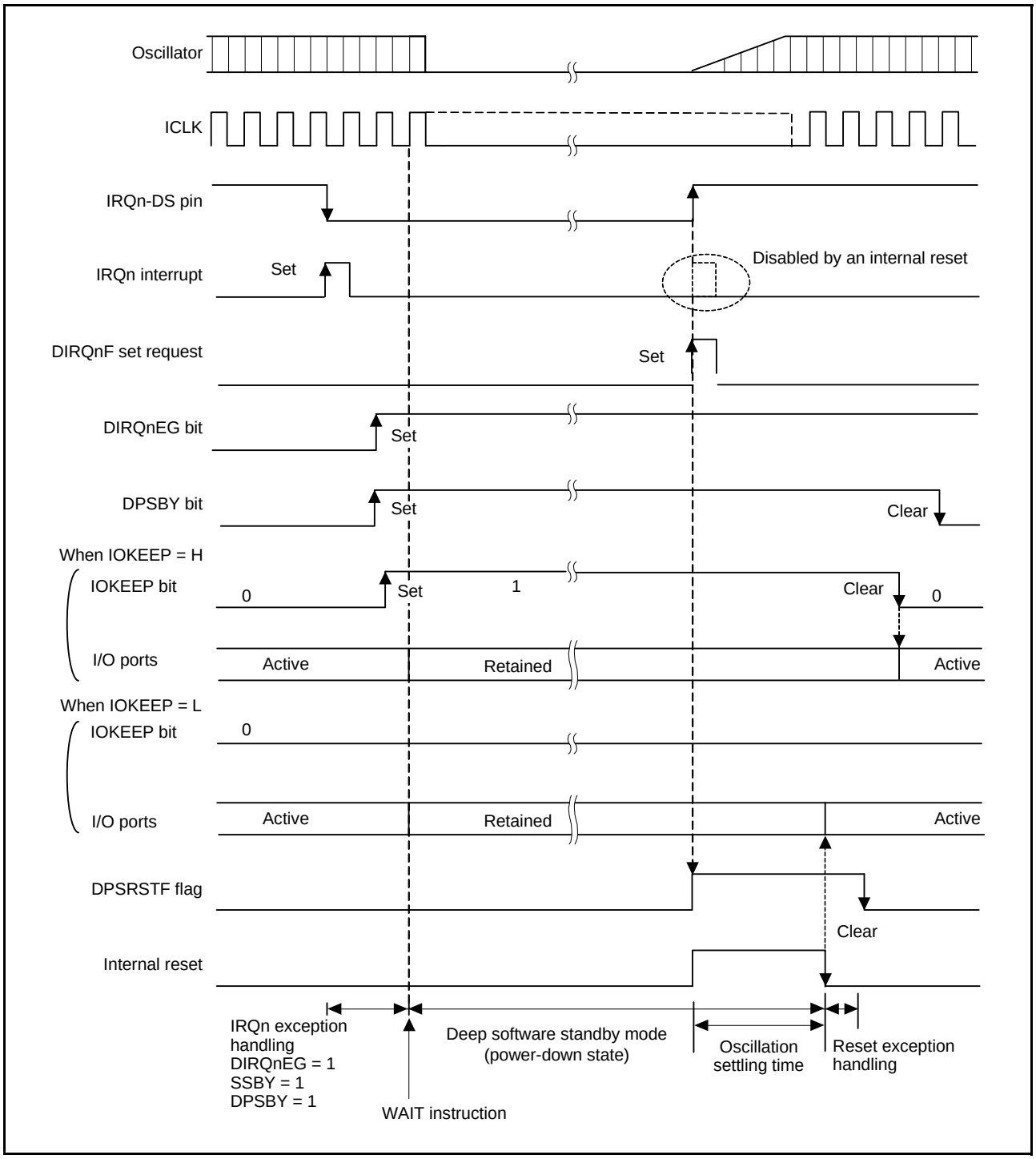


Figure 11.3 Example of Deep Software Standby Mode Application

11.6.4.5 Flowchart to Use Deep Software Standby Mode

Figure 11.4 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 handling to determine whether a reset was generated by the RES# pin or by the cancellation of 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 the cancellation of deep software standby mode, the DPSBYCR.IOKEEP bit is cleared to 0 after the I/O port settings have been made.

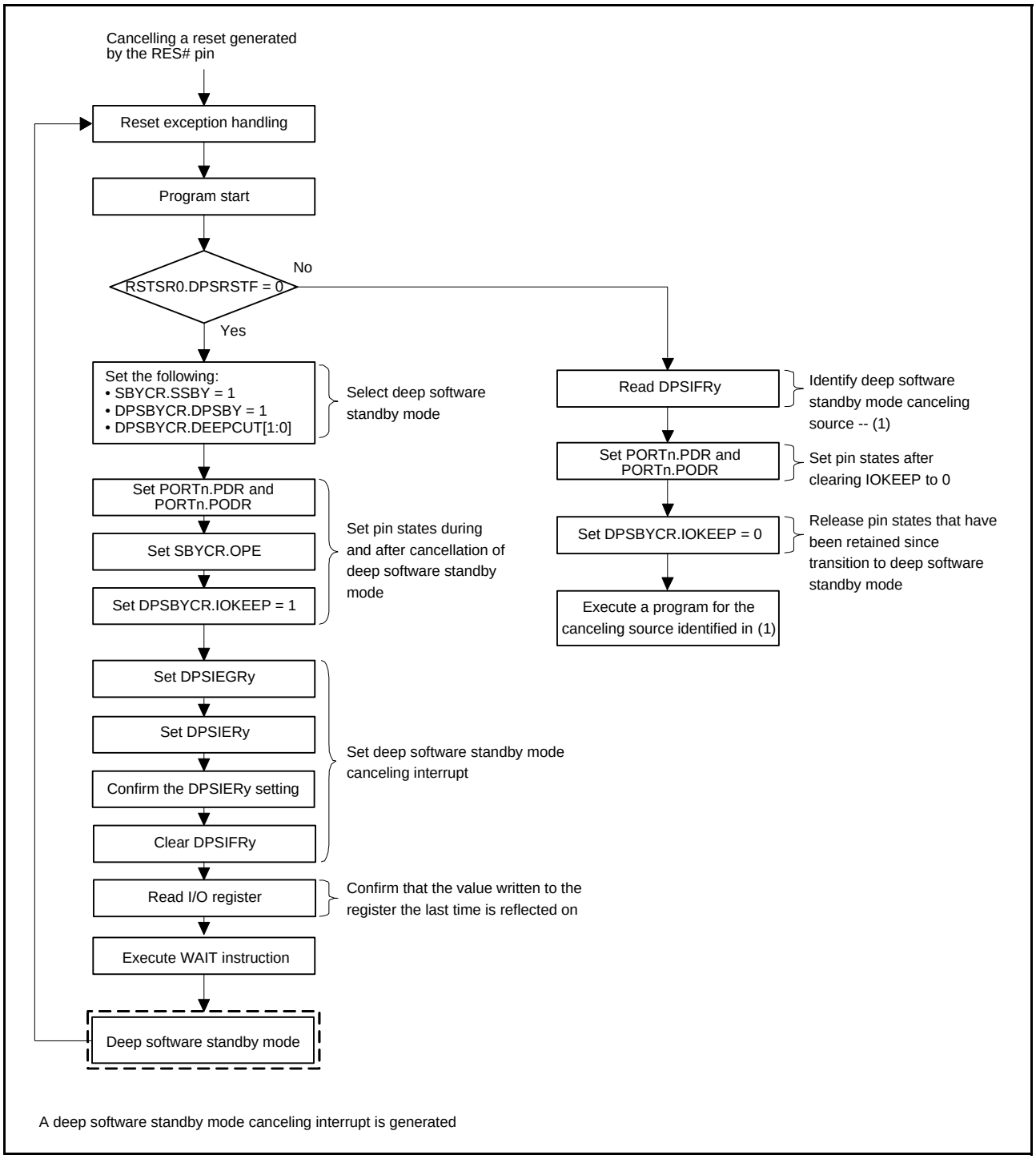


Figure 11.4 Example of Flowchart to Use Deep Software Standby Mode

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 17, DMA Controller (DMACA) and section 19, Data Transfer Controller (DTCa).

11.7.3 On-Chip Peripheral Module Interrupts

Interrupts do not operate in the module stop state. Therefore, if the module stop state is made 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, and MSTPCRC

Write accesses to MSTPCRA, MSTPCRB, and MSTPCRC 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

The WAIT instruction is executed before completion of the preceding register write. The WAIT instruction may be executed before the register setting modification is reflected, causing unintended operation. To avoid this, always execute the WAIT instruction after confirming that the last write to the register has completed.

11.7.7 Rewrite the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the DMAC 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 Transition to Software Standby Mode for Products with 1.5-Mbyte ROM or More or Products with 176 Pins or More

With the product with 1.5-Mbyte ROM or more or the product with the 176 pins or more, software standby mode is not entered even when a WAIT instruction is executed while the main clock oscillator or PLL circuit is selected as the clock source for the system clock. To enter software standby mode, select the low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), or sub-clock oscillator as the clock source before issuing the WAIT instruction.

11.7.9 Canceling All-Module Clock Stop Mode

If the ICLK is set so as to be slower than the PCLKA and PCLKB, a TMR interrupt cannot be used to cancel all-module clock stop mode. To use the TMR interrupt as the all-module clock stop mode cancelling source, change the ICLK so as to be faster than the PCLKA and PCLKB before all-module clock stop mode is entered.

11.7.10 Point for Caution when Using the Sub-Clock as the Source of the System Clock

If the sub-clock is in use as the source of the system clock, make sure that the RTC or the low-speed on-chip oscillator is operating (by setting the RCR3.RTCEN = 1 or the LOCOCR.LCSTP = 0, respectively) for a transition to software standby mode.

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.

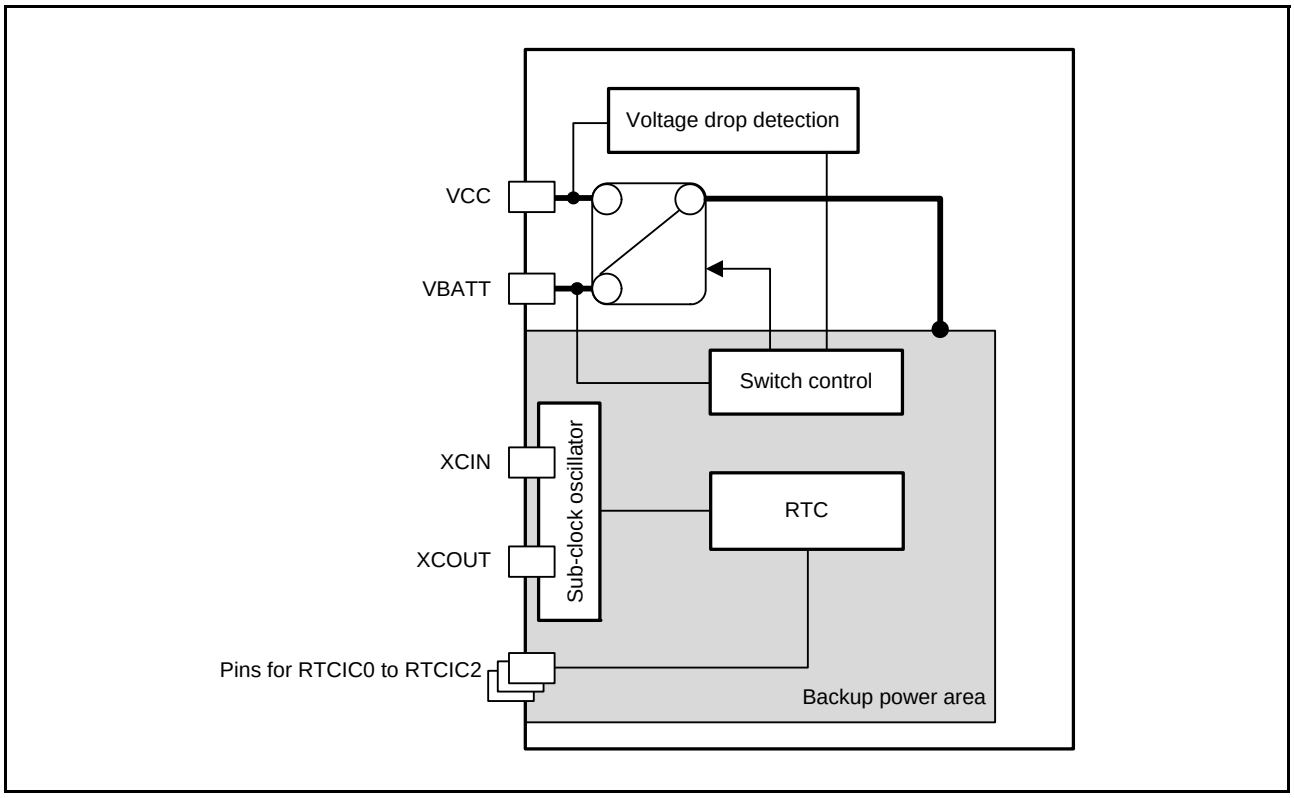


Figure 12.1 Configuration of Battery Backup Function

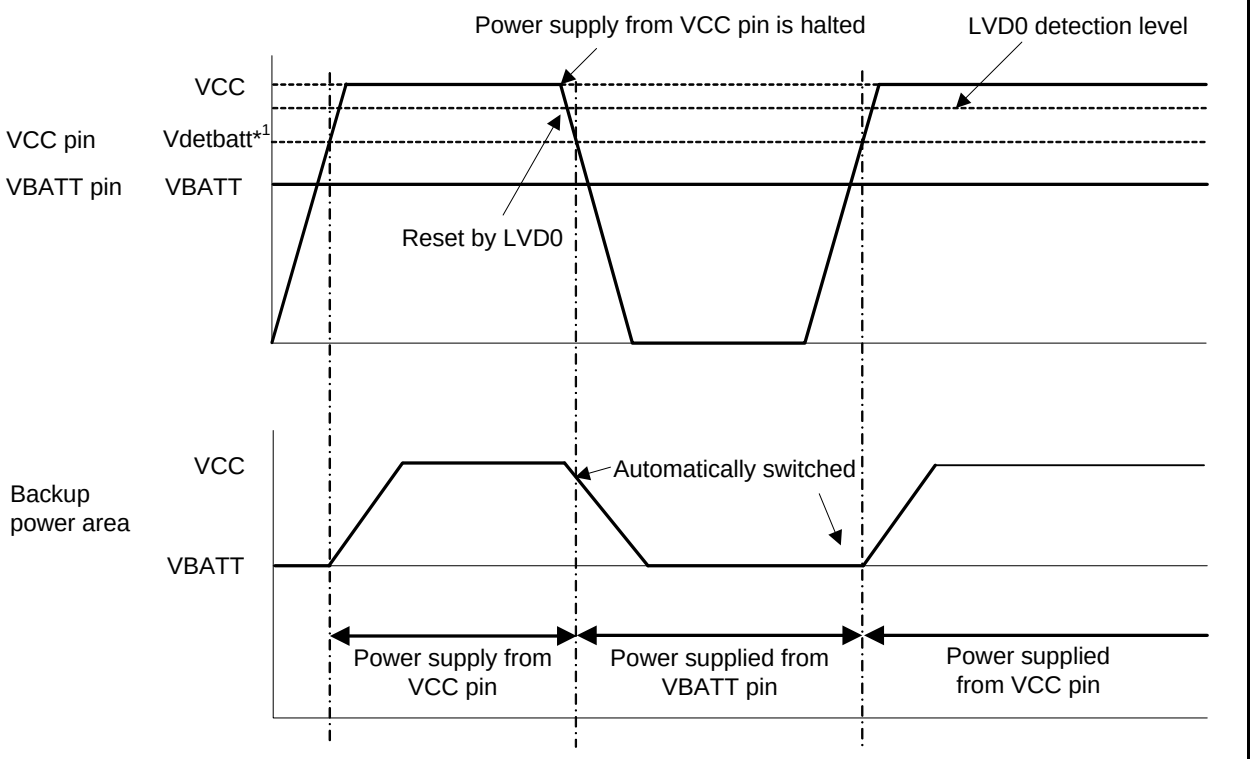
12.2 Operation

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)

Figure 12.2 shows the operation for switching to the battery backup function.



Note: • For details on electrical characteristics, see section 49, 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, OSTDCR, OSTDSR
PRC1	• Registers related to the operating modes: SYSCR0, SYSCR1 • Registers related to the low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, OPCCR, RSTCKCR, MOSCWTCR, SOSCWTCR, PLLWTCR, DPSBYCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR3 • Registers related to clock generation circuit: MOFCR, HOCOPCR • Software reset register: SWRR
PRC3	• Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR • Battery backup function: VBATTMNSELR

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 PRC0, PRC1, and PRC3 bits.	R/W

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.

PRKEY[7:0] Bits (PRC Key Code)

These bits enable or disable writing to PRCR. Before writing values to the PRCi bits (i = 0, 1, 3) in PRCR, set the PRKEY[7:0] bits to A5h. When the PRKEY[7:0] bits are set to a value other than A5h, the PRCi bits are not changed by a write operation.

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 RX CPU supports seven 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.

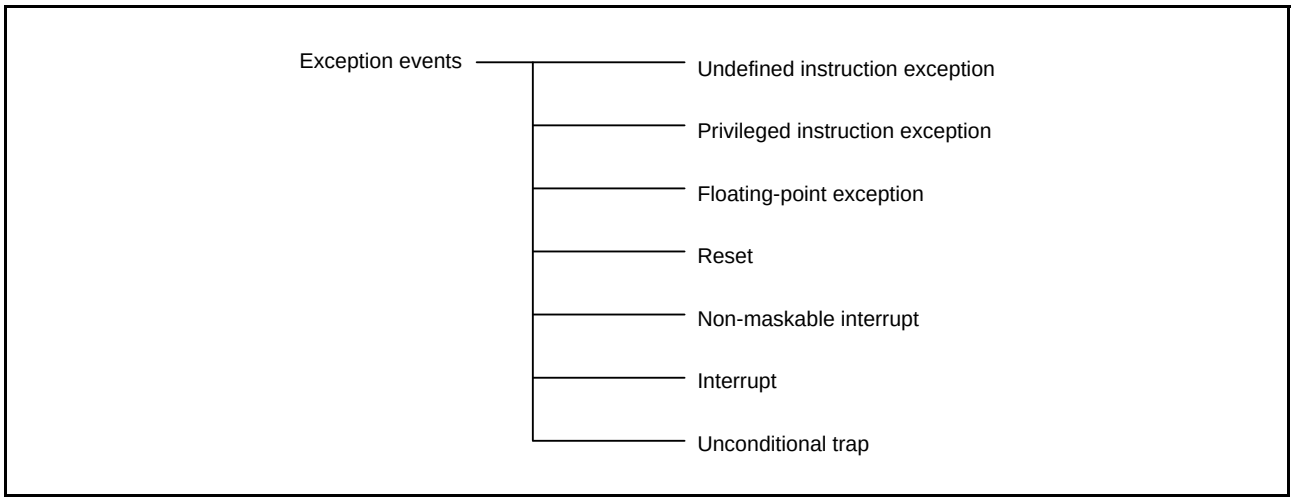


Figure 14.1 Types of Exception Events

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 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.

14.1.4 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.

14.1.5 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.

14.1.6 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.

14.1.7 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.

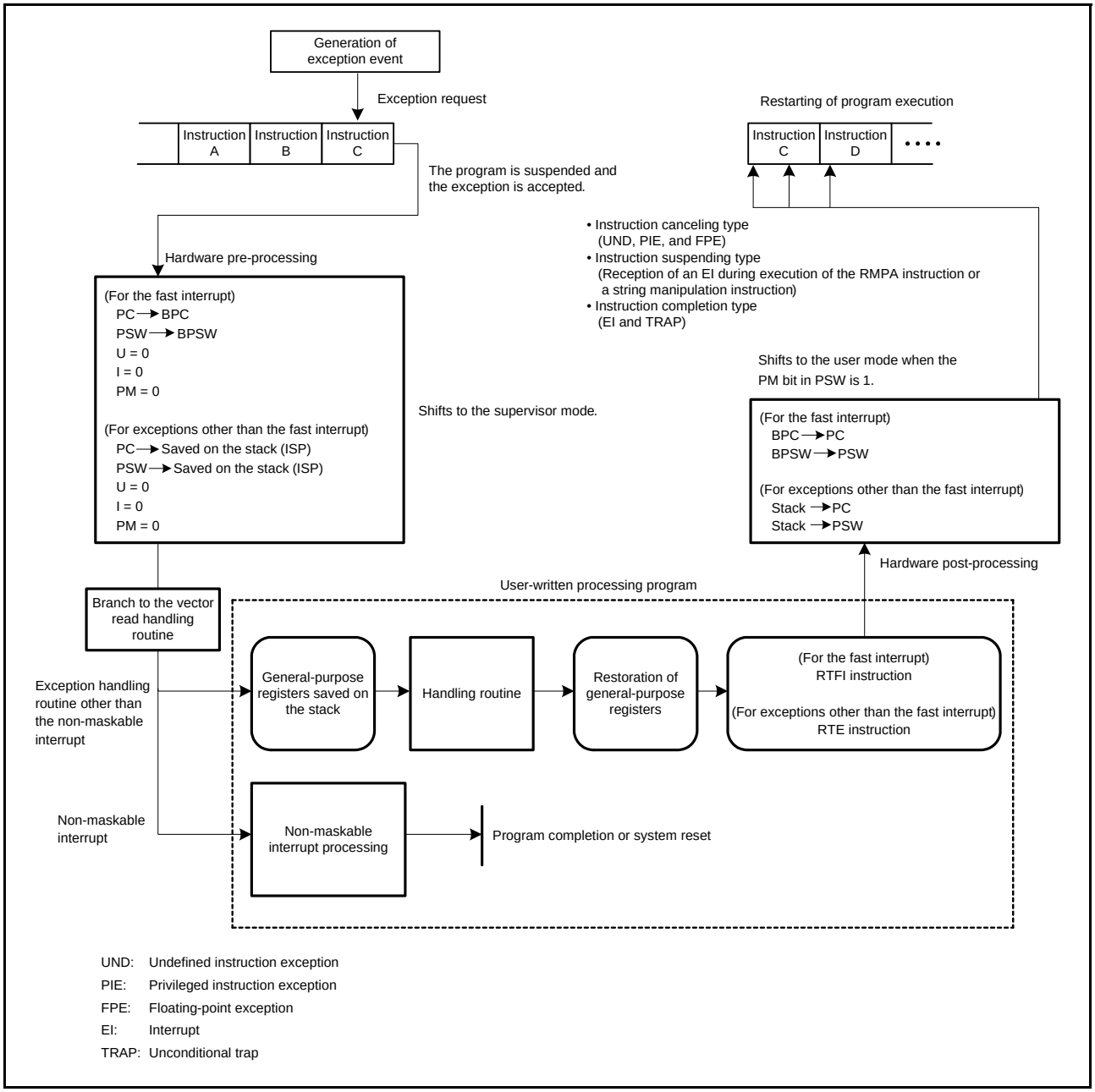


Figure 14.2 Outline of Exception Handling Procedure

When an exception is accepted, hardware processing by the RX 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 RX 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 RX 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
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.

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		Fixed vector table	Stack
Privileged instruction exception		Fixed vector table	Stack
Floating-point exception		Fixed vector table	Stack
Reset		Fixed vector table	Nowhere
Non-maskable interrupt		Fixed vector table	Stack
Interrupt	Fast interrupt	FINTV	BPC and BPSW
	Other than above	Relocatable vector table (INTB)	Stack
Unconditional trap		Relocatable 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 cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFDCh.
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 cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFD0h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.3 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 cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFE4h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.4 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.

14.5.5 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 cleared 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 address FFFF FFF8h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.6 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 cleared 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 relocatable 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.

14.5.7 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 cleared 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 relocatable vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the relocatable 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
Floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Return is impossible
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	Undefined instruction exception Privileged instruction exception
	5	Unconditional trap
Low	6	Floating-point exception

15. Interrupt Controller (ICUb)

15.1 Overview

The interrupt controller receives interrupt signals from the peripheral modules and external pins, sends interrupts to the CPU, and activates the DTC and DMAC.

Table 15.1 lists the specifications of the interrupt controller, and Figure 15.1 shows a block diagram of the interrupt controller.

Table 15.1 Specifications of Interrupt Controller

Item		Description
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules • Interrupt detection: Edge detection/level detection Edge detection or level detection is fixed for each source of connected peripheral modules. • Interrupt grouping: Multiple interrupt requests can be allocated to a single interrupt vector. Number of groups for edge detection interrupts: 7 (groups 0 to 6) Number of groups for level detection interrupts: 1 (group 12) • Interrupt unit selection: One of the two interrupt request units can be selected as an interrupt request source. Number of interrupts per unit: 6
	External pin interrupts	• Interrupts from pins IRQ0 to IRQ15 • Number of sources: 16 • Interrupt detection: Low level/falling edge/rising edge/rising and falling edges One of these detection methods can be set for each source. • Digital filter function: supported
	Software interrupt	• Interrupt generated by writing to a register • One interrupt source
	Interrupt priority	Specified by registers.
	Fast interrupt function	Faster interrupt processing of the CPU can be set only for a single interrupt source.
	DTC/DMAC control	The DTC and DMAC can be activated by interrupt sources.*1
Non-maskable interrupts	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection: Falling edge/rising edge • Digital filter function: supported
	Oscillation stop detection interrupt	Interrupt on detection of oscillation having stopped
	WDT underflow/ refresh error	Interrupt on underflow of the counter in interval-timer mode
	IWDT underflow/ refresh error	Interrupt on underflow of the down counter
	Voltage-monitoring 1 interrupt	Voltage monitoring interrupt of voltage detection circuit 1 (LVD1)
	Voltage-monitoring 2 interrupt	Voltage monitoring interrupt of voltage detection circuit 2 (LVD2)
Return from power-down modes		• Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source. • All-module clock stop mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ15 interrupts, TMR interrupts, USB resume interrupts, or RTC alarm/ periodic interrupts. • Software standby mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ15 interrupts, USB resume interrupts, or RTC alarm/periodic interrupts.

Note 1. For the DTC and DMAC activation sources, refer to Table 15.3, Interrupt Vector Table.

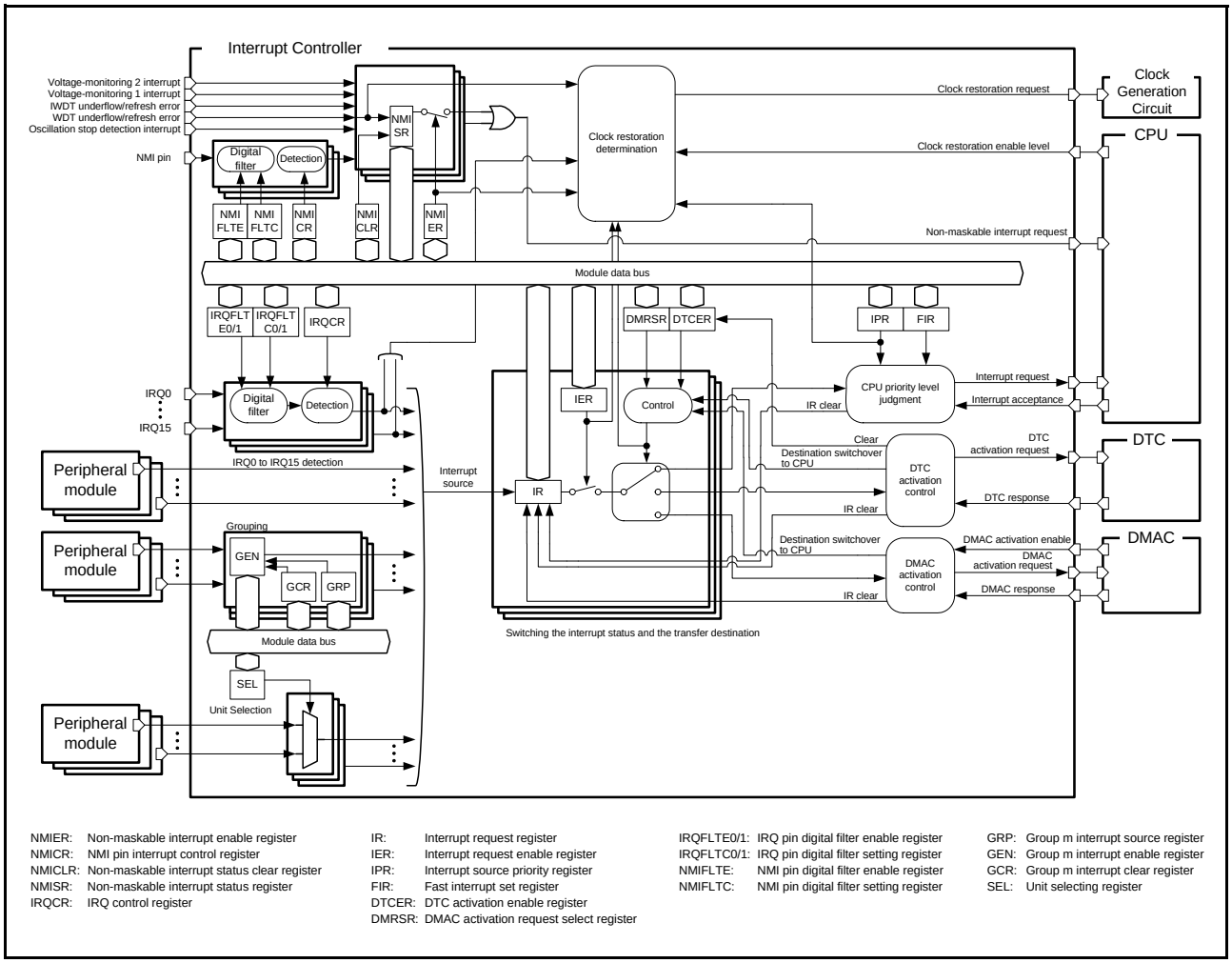


Figure 15.1 Block Diagram of Interrupt Controller

Table 15.2 lists the input/output pins of the interrupt controller.

Table 15.2 Pin Configuration of Interrupt Controller

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 = interrupt vector number)

Address(es): 0008 7010h to 0008 70FDh

	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	These bits are read as 0. The write value should be 0.	R/W

Note 1. For an edge detection interrupt, only 0 can be written to this bit; do not write 1.
For a level detection interrupt, neither 0 nor 1 can be written.

IRn is provided for each interrupt source, where “n” indicates the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, see Table 15.3, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

This bit is the status flag of an individual interrupt request. This flag is set to 1 when the corresponding interrupt request is generated. To detect an interrupt request, the interrupt request output should be enabled by the corresponding peripheral module interrupt enable bit.
There are two interrupt request detection methods: edge detection and level detection. For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source. For interrupts from IRQi pins, edge detection or level detection is selected by the setting of the IRQMD[1:0] bits in the corresponding IRQCRi (i = 0 to 15).
For detection of the various interrupt sources, see Table 15.3, Interrupt Vector Table.
Grouped interrupt requests are detected using group m interrupt source register (GRPm; m = group number). The interrupt request detected by GRPm is further detected as a level detection interrupt request by the IR flag corresponding to the group. For details of the interrupt grouping function, see section 15.4, Peripheral Module Interrupt Request Groups and Unit Selection.
For grouped interrupt requests, see Table 15.4, Group m Interrupt Requests.

(1) Edge detection

[Setting condition]

- The flag is set to 1 in response to the generation of an interrupt request from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing conditions]

- The flag is cleared to 0 when the interrupt request destination accepts the interrupt request.
- The IR flag is cleared to 0 by writing 0 to it. Note, however, that writing 0 to the IR flag is prohibited if the destination of the interrupt request is the DTC or DMAC.

(2) Level detection

[Setting conditions]

- The flag remains set to 1 while an interrupt request is being sent from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.
- For grouped interrupt requests, the flag is set to 1 when both the interrupt request enable bit j in group m interrupt enable register (GENm.ENj; m = group number and j = bit number) and interrupt status flag in group m interrupt source register (GRPm.ISj) are 1.

[Clearing conditions]

- The flag is cleared to 0 when the source of the interrupt request is cleared (it is not cleared when the interrupt request destination accepts the interrupt request). For clearing interrupts from the various peripheral modules, refer to the sections describing the modules.
- For grouped interrupt requests, the flag is cleared to 0 when either the GENm.ENj bit or GRPm.ISj flag is 0.

When level detection has been selected for an IRQi pin, the interrupt request is withdrawn by driving the IRQi pin high. Do not write 0 or 1 to the IR flag while level detection is selected.

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: • Write 0 to the bit that corresponds to the vector number for reservation. These bits are read as 0.

IENj Bits (Interrupt Request Enable j) (j = 7 to 0)

When an IENj bit is 1, the corresponding interrupt request will be output to the destination selected for the request.
When an IENj bit is 0, the corresponding interrupt request will not be output to the destination selected for the request.
The setting of an IENj bit does not affect the IRn.IR flag. Even if the corresponding IENj bit is 0, the IR flag value changes according to the descriptions in section 15.2.1, Interrupt Request Register n (IRn) (n = interrupt vector number).
The IERm.IENj bit is set for each request source (vector number).
For the correspondence between interrupt sources and IERm.IENj bits, see Table 15.3, Interrupt Vector Table.
For the procedure for setting IERm.IENj bits during the selection of destinations for interrupt requests, refer to section 15.5.3, Selecting Interrupt Request Destinations.

15.2.3 Interrupt Source Priority Register n (IPRn) (n = 000 to 253)

Address(es): 0008 7300h to 0008 73FDh

	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 prohibited)*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	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the interrupt is specified as a fast interrupt, it can be issued even if the priority level is level 0.

For the correspondence between interrupt sources and IPRn registers, see Table 15.3, Interrupt Vector Table.

IPR[3:0] Bits (Interrupt Priority Level Select)

These bits specify the priority level of the corresponding interrupt source.
Priority levels specified by the IPR[3:0] bits are used only to determine the priority of interrupt requests to be transferred to the CPU, and do not affect activation requests to the DTC and DMAC.
The CPU accepts only interrupt requests higher than the priority level specified by the IPL[3:0] bits in PSW, and handles accepted interrupts.
If two or more interrupt requests are generated at the same time, their priority levels are compared with the value of the IPR[3:0] bits. If interrupt requests of the same priority level are generated at the same time, an interrupt source with a smaller vector number takes precedence.
These bits should be written to while an interrupt request is disabled (the IERm.IENj bit = 0).

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	Specify the vector number of an interrupt source to be a fast interrupt.	R/W
b14 to b8	—	Reserved	These bits are read as 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 fast interrupt function based on the FIR register setting is applicable only to interrupts to the CPU. It will not affect any transfer request to the DTC or DMAC.
Before writing to this register, be sure to disable interrupt requests (IERm.IENj bit = 0).

FVCT[7:0] Bits (Fast Interrupt Vector Number)

The FVCT[7:0] bits specify the vector number of an interrupt source that uses the fast interrupt function.

FIEN Bit (Fast Interrupt Enable)

This bit enables fast interrupt.
Setting this bit to 1 makes the interrupt request of the vector number specified by the FVCT[7:0] bits a fast interrupt. When an interrupt request of the vector number specified by the FVCT[7:0] bits is generated and the interrupt request destination is the CPU while the FIEN bit is 1, the interrupt request is output to the CPU as a fast interrupt regardless of the setting of the IPRn register. When using the fast interrupt for returning from the software standby mode, see section 15.7.3, Return from Software Standby Mode.
If the setting of the IERm.IENj (m = 02h to 1Fh, j = 7 to 0) bit has disabled interrupt requests from the interrupt source with the vector number in this register, fast interrupt requests are not output to the CPU.
For settable vector numbers, see Table 15.3, Interrupt Vector Table.
Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details on fast interrupt, see section 14, Exception Handling, and section 15.5.5, Fast Interrupt.

15.2.5 Software Interrupt Activation 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 Activation	This bit is read as 0. Writing 1 issues a software interrupt request. Writing 0 to this bit has no effect.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written.

SWINT Bit (Software Interrupt Activation)

When 1 is written to the SWINT bit, the interrupt request register 027 (IR027) is set to 1.
If 1 is written to the SWINT bit when the DTC activation enable register 027 (DTCER027) is set to 0, an interrupt to the CPU is generated.
If 1 is written to the SWINT bit when the DTC activation enable register 027 (DTCER027) is set to 1, a DTC activation request is issued.

15.2.6 DTC Activation Enable Register n (DTCERn) (n = interrupt vector number)

Address(es): 0008 711Bh to 0008 71FBh

	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 Activation Enable	0: DTC activation is disabled 1: DTC activation is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

An interrupt source that has been selected as a source for DMAC activation should not be specified as a source for DTC activation. See Table 15.3, Interrupt Vector Table, for the interrupt sources that are selectable as sources for DTC activation.

DTCE Bit (DTC Activation Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the source for the DTC activation.

[Setting condition]

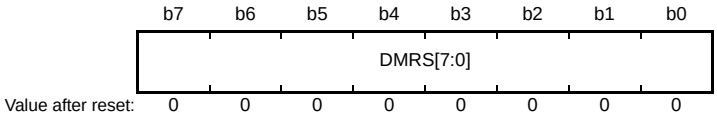
- When 1 is written to the DTCE bit

[Clearing conditions]

- When the specified number of transfers is completed (for the chain transfer, the number of transfers for the last chain transfer is completed)
- When 0 is written to the DTCE bit

15.2.7 DMAC Activation Request Select Register m (DMRSRm) (m = DMAC channel number)

Address(es): DMRSR0 0008 7400h, DMRSR1 0008 7404h
DMRSR2 0008 7408h, DMRSR3 0008 740Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DMRS[7:0]	DMAC Activation Source Select	These bits specify the vector number for the DMAC activation request.	R/W

To specify the same interrupt source for multiple DMRSRm registers is disabled. The interrupt source that has been selected for the DMRSRm activation should not be specified as the source for the DTC activation. Otherwise, the correct operation is not guaranteed.

DMRS[7:0] Bits (DMAC Activation Source Select)

The vector number of the interrupt source for DMAC activation is specified in 8 bits. Do not set the vector numbers that are not assigned for the DMAC activation.

For the correspondence between interrupt sources and interrupt vector numbers, see Table 15.3, Interrupt Vector Table.

Write to the DMRSRm register while the DMA transfer enable bit of the DMA transfer enable register (DMACm.DMCNT.DTE) is cleared to 0.

15.2.8 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	These bits are read as 0. The write value should be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Sense 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	These bits are read as 0. The write value should be 0.	R/W

Only change the settings of this register while the corresponding interrupt request enable bit is prohibiting the interrupt request (IENj bit in IERm is 0). After changing the setting, clear the IR flag in IRn before setting the interrupt enable bit. However, when the change is to the low level, the IR flag does not require clearing.

IRQMD[1:0] Bits (IRQ Detection Sense Select)

These bits select the detection sensing method of external pin interrupt sources IRQ0 to IRQ15.
For the external pin interrupt detection setting, see section 15.5.7, External Pin Interrupts.

15.2.9 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): 0008 7510h

	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

FLTENi Bits (IRQi Digital Filter Enable) (i = 0 to 7)

These bits enable the digital filter used for the external pin interrupt sources IRQ0 to IRQ7.

The digital filter is enabled when the FLTENi bit is 1, and disabled when the FLTENi bit is 0.

The IRQi pin level is sampled at the sampling clock cycle specified with the IRQFLTC0.FCLKSELi[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, see section 15.5.6, Digital Filter.

15.2.10 IRQ Pin Digital Filter Enable Register 1 (IRQFLTE1)

Address(es): 0008 7511h

	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

FLTENi Bits (IRQi Digital Filter Enable) (i = 8 to 15)

These bits enable the digital filter used for the external pin interrupt sources IRQ8 to IRQ15.

The digital filter is enabled when the FLTENi bit is 1, and disabled when the FLTENi bit is 0.

The IRQi pin level is sampled at the sampling clock cycle specified with the IRQFLTC1.FCLKSELi[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, see section 15.5.6, Digital Filter.

15.2.11 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): 0008 7514h

	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: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/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

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0 to 7)

These bits select the cycle of the digital filter sampling clock for the external pin interrupt request pins IRQ0 to IRQ7. The sampling clock cycle can be selected from among the PCLK (every cycle), PCKL/8 (once every eight cycles), PCKL/32 (once every 32 cycles), and PCKL/64 (once every 64 cycles).
For details of the digital filter, see section 15.5.6, Digital Filter.

15.2.12 IRQ Pin Digital Filter Setting Register 1 (IRQFLTC1)

Address(es): 0008 7516h

	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: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/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

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 8 to 15)

These bits select the cycle of the digital filter sampling clock for the external pin interrupt request pins IRQ8 to IRQ15. The sampling clock cycle can be selected from among the PCLK (every cycle), PCKL/8 (once every eight cycles), PCKL/32 (once every 32 cycles), and PCKL/64 (once every 64 cycles).
For details of the digital filter, see section 15.5.6, Digital Filter.

15.2.13 Non-Maskable Interrupt Status Register (NMISR)

Address(es): 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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
b7, b6	—	Reserved	These bits are read as 0 and cannot be modified.	R

The NMISR register monitors the status of a non-maskable interrupt source. Writing to the NMISR register is ignored. The setting in the non-maskable interrupt enable register (NMIER) does not affect the status flags in NMISR.
Before the end of the non-maskable interrupt handler, read the NMISR register and confirm the generation status of other non-maskable interrupts. Be sure to confirm that all of the bits in the NMISR register are set to 0 before the end of the handler.

NMIST Flag (NMI Status Flag)

This flag indicates the NMI pin interrupt request.
The NMIST flag is read-only, and cleared by the NMICLR.NMICLR bit.
[Setting condition]

- When an edge specified by the NMICR.NMIMD bit is input to the NMI pin

[Clearing condition]

- When 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

This flag indicates the oscillation stop detection interrupt request.
The OSTST flag is read-only, and cleared by the NMICLR.OSTCLR bit.
[Setting condition]

- When the oscillation stop detection interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.OSTCLR bit

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

This flag indicates the WDT underflow/refresh error interrupt request.
The WDTST flag is read-only, and cleared by the NMICLR.WDTCLR bit.
[Setting condition]

- When the WDT underflow/refresh error interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.WDTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

This flag indicates the IWDT underflow/refresh error interrupt request.
The IWDTST flag is read-only, and cleared by the NMICLR.IWDTCLR bit.
[Setting condition]

- When the IWDT underflow/refresh error interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage-Monitoring 1 Interrupt Status Flag)

This flag indicates the request for voltage-monitoring 1 interrupt.
The LVD1ST flag is read-only, and cleared by the NMICLR.LVD1CLR bit.
[Setting condition]

- When the voltage-monitoring 1 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD1CLR bit

LVD2ST Flag (Voltage-Monitoring 2 Interrupt Status Flag)

This flag indicates the request for voltage-monitoring 2 interrupt.
The LVD2ST flag is read-only, and cleared by the NMICLR.LVD2CLR bit.
[Setting condition]

- When the voltage-monitoring 2 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD2CLR bit

15.2.14 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

NMIEN Bit (NMI Pin Interrupt Enable)

This bit enables the NMI pin interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

This bit enables the oscillation stop detection interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

WDTEN Bit (WDT Underflow/Refresh Error Enable)

This bit enables the WDT underflow/refresh error interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

This bit enables the IWDT underflow/refresh error interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

LVD1EN Bit (Voltage-Monitoring 1 Interrupt Enable)

This bit enables the voltage-monitoring 1 interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

LVD2EN Bit (Voltage-Monitoring 2 Interrupt Enable)

This bit enables the voltage-monitoring 2 interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

15.2.15 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	This bit is read as 0. Writing 1 to this bit clears the NMISR.NMIST flag. Writing 0 to this bit has no effect.	R/(W) *1
b1	OSTCLR	OST Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.OSTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b2	WDTCLR	WDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.WDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b3	IWDTCLR	IWDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.IWDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b4	LVD1CLR	LVD1 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD1ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b5	LVD2CLR	LVD2 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD2ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b7, b6	—	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.

NMICLR Bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag.

The 1 state is not retained. This bit is read as 0.

OSTCLR Bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag.

The 1 state is not retained. This bit is read as 0.

WDTCLR Bit (WDT Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag.

The 1 state is not retained. This bit is read as 0.

IWDTCLR Bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag.

The 1 state is not retained. This bit is read as 0.

LVD1CLR Bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag.

The 1 state is not retained. This bit is read as 0.

LVD2CLR Bit (LVD2 Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag.
The 1 state is not retained. This bit is read as 0.

15.2.16 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	These bits are read as 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	These bits are read as 0 and cannot be modified.	R/W

Change the setting of the NMICR register before the NMI pin interrupt is enabled (before setting the NMIER.NMIEN bit to 1).

15.2.17 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	These bits are read as 0. The write value should be 0.	R/W

NFLTEN Bit (NMI Digital Filter Enable)

This bit enables the digital filter used for the NMI pin interrupt.
The digital filter is enabled when the NFLTEN bit is 1, and disabled when the NFLTEN bit is 0.
The NMI pin level is sampled at the sampling clock cycle specified with the NMIFLTC.NFCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.
For details of the digital filter, see section 15.5.6, Digital Filter.

15.2.18 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: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/64	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFCLKSEL[1:0] Bits (NMI Digital Filter Sampling Clock)

These bits select the cycle of the digital filter sampling clock for the NMI pin interrupt.

The sampling clock cycle can be selected from among the PCLK (every cycle), PCKL/8 (once every eight cycles), PCKL/32 (once every 32 cycles), and PCKL/64 (once every 64 cycles).

For details of the digital filter, see section 15.5.6, Digital Filter.

15.2.19 Group m Interrupt Source Register (GRPm) (m: group number)

- GRP00, GRP01, GRP02

Address(es): GRP00 0008 C300h, GRP01 0008 C304h, GRP02 0008 C308h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	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
b31 to b3	—	Reserved	These bits are read as 0 and cannot be modified.	R

- GRP03, GRP04, GRP05, GRP06

Address(es): GRP03 0008 C30Ch, GRP04 0008 C310h, GRP05 0008 C314h, GRP06 0008 C318h

	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
	—	—	—	—	—	—	—	—	—	—	—	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
b31 to b5	—	Reserved	These bits are read as 0 and cannot be modified.	R

- GRP12

Address(es): GRP12 0008 C330h

	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
	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
b31 to b16	—	Reserved	These bits are read as 0 and cannot be modified.	R

GRPm is provided for each group, where m indicates the group number (m: 00 to 06 or 12).
For details of the grouping function, see section 15.4.1, Interrupt Request Groups. For correspondence between the interrupt request allocated to each group and the bit number in the GRPm register, see Figure 15.4, Group m Interrupt Requests. For correspondence between grouped interrupt sources and interrupt vector numbers, see Table 15.3, Interrupt Vector Table.

ISj Flag (Interrupt Status Flag j) (j = bit number)

This bit is the status flag of the peripheral module interrupt request allocated to the jth bit in group m. This flag is set to 1 when the interrupt request is generated and is enabled by the interrupt request enable bit j in group m interrupt enable register (GENm.ENj; m = group number and j = bit number).
When both the GENm.ENj bit and ISj flag bit for any of the interrupt sources in a group are 1, the interrupt status flag in interrupt request register n (IRn.IR; n = interrupt vector number) corresponding to the group is set to 1.
Groups 0 to 6 comprise edge detection interrupt requests and group 12 comprises level detection interrupt requests.

(1) Groups 0 to 6
[Setting condition]

- The ISj flag is set to 1 when the corresponding peripheral module interrupt request is generated and GENm.ENj is 1.

[Clearing condition]

- The ISj flag is cleared to 0 by writing 1 to the interrupt source clear bit in group m interrupt clear register (GCRm.CLRj).

(2) Group 12

[Setting condition]

- The ISj flag remains 1 while the corresponding peripheral module interrupt request is being sent and GENm.ENj is 1.

[Clearing conditions]

- The ISj flag is cleared to 0 when the source of the interrupt request is cleared.
- The ISj flag is cleared to 0 when GENm.ENj is 0.

Writing to the ISj flag is disabled.

15.2.20 Group m Interrupt Enable Register (GENm) (m = group number)

- GEN00, GEN01, GEN02

Address(es): GEN00 0008 C340h, GEN01 0008 C344h, GEN02 0008 C348h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	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
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- GEN03, GEN04, GEN05, GEN06

Address(es): GEN03 0008 C34Ch, GEN04 0008 C350h, GEN05 0008 C354h, GEN06 0008 C358h

	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
	—	—	—	—	—	—	—	—	—	—	—	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
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- GEN12

Address(es): GEN12 0008 C370h

	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
	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
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

For details of the grouping function, see section 15.4.1, Interrupt Request Groups.
GENm is provided for each group, where m indicates the group number (m: 00 to 06 or 12).

ENj Bits (Interrupt Request Enable j) (j = bit number)

- Groups 00 to 06

When an ENj bit is 1, interrupt request detection by the corresponding GRPm.ISj flag (m = group number and j = bit number) is enabled. When an interrupt is detected, the GRPm.ISj flag is set to 1.
When an ENj bit is 0, interrupt request detection is disabled.
Writing 0 to the ENj bit does not affect the corresponding GRPm.ISj flag.

- Group 12

When an ENj bit is 1, interrupt request detection by the corresponding GRPm.ISj flag (m = group number and j = bit number) is enabled. When an interrupt is detected, the GRPm.ISj flag is set to 1.
When an ENj bit is 0, interrupt request detection is disabled.
Writing 0 to the ENj bit does not affect the corresponding GRPm.ISj flag.

15.2.21 Group m Interrupt Clear Register (GCRm) (m = group number)

- GCR00, GCR01, GCR02

Address(es): GCR00 0008 C380h, GCR01 0008 C384h, GCR02 0008 C388h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	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	These bits are read as 0. Writing 1 to this bit clears the interrupt status flag bit (GRPm.ISj) of the same group and bit. Writing 0 is invalid.	R(/W) *1
b1	CLR1	Interrupt source clear 1		R(/W) *1
b2	CLR2	Interrupt source clear 2		R(/W) *1
b31 to b3	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note 1. Only 1 can be written.

- GCR03, GCR04, GCR05, GCR06

Address(es): GCR03 0008 C38Ch, GCR04 0008 C390h, GCR05 0008 C394h, GCR06 0008 C398h

	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
	—	—	—	—	—	—	—	—	—	—	—	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	Writing 1 to this bit clears the interrupt status flag bit (GRPm.ISj) of the same group and bit. Writing 0 is invalid.	R(/W) *1
b1	CLR1	Interrupt source clear 1		R(/W) *1
b2	CLR2	Interrupt source clear 2		R(/W) *1
b3	CLR3	Interrupt source clear 3		R(/W) *1
b4	CLR4	Interrupt source clear 4		R(/W) *1
b31 to b5	—	Reserved	These bits are read as 0 and cannot be modified.	R/W

Note 1. Only 1 can be written.

- GCR12

Address(es): GCR12 0008 C3B0h

	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
	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	Writing 1 to this bit clears the interrupt status flag bit (GRPm.ISj) of the same group and bit. Writing 0 is invalid.	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)
b31 to b16	—	Reserved	These bits are read as 0 and cannot be modified.	R/W

For details of the grouping function, see section 15.4.1, Interrupt Request Groups.
GCRm is provided for each group, where m indicates the group number (m: 00 to 06, or 12).

CLRj Bits (Interrupt Source Clear j) (j = bit number)

Writing 1 to this bit clears the corresponding GRPm.ISj flag (m = group number and j = bit number) to 0.

15.2.22 Unit Selecting Register (SEL)

Address(es): 0008 C3C0h

	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
	—	—	—	—	—	—	—	—	—	—	CN5	CN4	CN3	CN2	CN1	CN0
Value after 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	CN0	Unit Select 0	0: MTU0 interrupt requests are selected. 1: TPU6 interrupt requests are selected.	R/W
b1	CN1	Unit Select 1	0: MTU1 interrupt requests are selected. 1: TPU7 interrupt requests are selected.	R/W
b2	CN2	Unit Select 2	0: MTU2 interrupt requests are selected. 1: TPU8 interrupt requests are selected.	R/W
b3	CN3	Unit Select 3	0: MTU3 interrupt requests are selected. 1: TPU9 interrupt requests are selected.	R/W
b4	CN4	Unit Select 4	0: MTU4 interrupt requests are selected. 1: TPU10 interrupt requests are selected.	R/W
b5	CN5	Unit Select 5	0: MTU5 interrupt requests are selected. 1: TPU11 interrupt requests are selected.	R/W
b6	—	Reserved	This bit is read as 0 and cannot be modified.	R

For details of the unit selection function and MTUn (n = 0 to 5) and TPUn (n = 6 to 11) interrupt sources, see section 15.4.2, Unit Selection.

CNj Bits (Unit Select j) (j = 0 to 5)

When a CNj bit is 0, the MTUn (n = 0 to 5) is selected. When a CNj bit is 1, the TPUn (n = 6 to 11) is selected.
Set the CNj bits before generating an interrupt request from the peripheral module of the selected unit; otherwise, operation cannot be guaranteed.

15.3 Vector Table

There are two types of interrupts detected by the interrupt controller: maskable interrupts and non-maskable interrupts. When the CPU accepts an interrupt or non-maskable interrupt, it acquires a four-byte vector address from the vector table.

15.3.1 Interrupt Vector Table

The interrupt vector table is placed in the 1024-byte range (4 bytes x 256 sources) beginning at the address specified in the interrupt table register (INTB) of the CPU. Write a value to the INTB register before enabling interrupts. The value written to the INTB register should be a multiple of 4.
Table 15.3 lists details of the interrupt vectors. Details of the headings in Table 15.3 are listed below.

Item	Description
177/176-pin package	"○" in these columns in Table 15.3 indicates that the interrupts are available with the package (the number of pins).
145/144-pin package	
100-pin package	
80-pin package	
Source of interrupt request generation	Name of the source for generation of the interrupt request
Name	Name of the interrupt
Vector no.	Vector number for the interrupt
Vector address offset	Value of the offset from the base address for the vector table
Form of interrupt detection	"Edge" or "level" as the method for detection of the interrupt
CPU interrupt	"○" in this column indicates usability as a CPU interrupt.
DTC activation	"○" in this column indicates usability as a request for DTC activation.
DMAC activation	"○" in this column indicates usability as a request for DMAC activation.
sstb return	"○" in this column indicates usability as a request for return from software-standby mode.
sacs return	"○" in this column indicates usability as a request for return from all-module clock-stop mode.
IER	Name of the interrupt request enable register (IER) and bit corresponding to the vector number
IPR	Name of the interrupt source priority register (IPR) corresponding to the interrupt source
DTCER	Name of the DTC activation enable register (DTCER) corresponding to the DTC activation source

Table 15.3 Interrupt Vector Table (1/7)

177/176-pin package	145/144-pin package	100-pin package	Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
—	—	—	—	Reserved	0	0000h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	1	0004h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	2	0008h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	3	000Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	4	0010h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	5	0014h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	6	0018h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	7	001Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	8	0020h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	9	0024h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	10	0028h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	11	002Ch	—	×	×	×	×	×	—	—	—

Table 15.3 Interrupt Vector Table (2/7)

177/176-pin package	145/144-pin package	100-pin package	Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
—	—	—	—	Reserved	12	0030h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	13	0034h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	14	0038h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	15	003Ch	—	×	×	×	×	×	—	—	—
○	○	○	BSC	BUSERR	16	0040h	Level	○	×	×	×	×	IER02.IEN0	IPR000	—
—	—	—	—	Reserved	17	0044h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	18	0048h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	19	004Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	20	0050h	—	×	×	×	×	×	—	—	—
○	○	○	FCU	FIFERR	21	0054h	Level	○	×	×	×	×	IER02.IEN5	IPR001	—
—	—	—	—	Reserved	22	0058h	—	×	×	×	×	×	—	—	—
○	○	○	FCU	FRDYI	23	005Ch	Edge	×	×	×	×	×	IER02.IEN7	IPR002	—
—	—	—	—	Reserved	24	0060h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	25	0064h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	26	0068h	—	×	×	×	×	×	—	—	—
○	○	○	ICU	SWINT	27	006Ch	Edge	○	○	×	×	×	IER03.IEN3	IPR003	DTCER027
○	○	○	CMT0	CMI0	28	0070h	Edge	○	○	○	×	×	IER03.IEN4	IPR004	DTCER028
○	○	○	CMT1	CMI1	29	0074h	Edge	○	○	○	×	×	IER03.IEN5	IPR005	DTCER029
○	○	○	CMT2	CMI2	30	0078h	Edge	○	○	○	×	×	IER03.IEN6	IPR006	DTCER030
○	○	○	CMT3	CMI3	31	007Ch	Edge	○	○	○	×	×	IER03.IEN7	IPR007	DTCER031
○	○	○	Ether	EINT	32	0080h	Level	○	×	×	×	×	IER04.IEN0	IPR032	—
○	○	○	USB0	D0FIFO0	33	0084h	Edge	○	○	○	×	×	IER04.IEN1	IPR033	DTCER033
○	○	○		D1FIFO0	34	0088h	Edge	○	○	○	×	×	IER04.IEN2	IPR034	DTCER034
○	○	○		USBI0	35	008Ch	Edge	○	×	×	×	×	IER04.IEN3	IPR035	—
○	—	—	USB1	D0FIFO1	36	0090h	Edge	○	○	○	×	×	IER04.IEN4	IPR036	DTCER036
○	—	—		D1FIFO1	37	0094h	Edge	○	○	○	×	×	IER04.IEN5	IPR037	DTCER037
○	—	—		USBI1	38	0098h	Edge	○	×	×	×	×	IER04.IEN6	IPR038	—
○	○	○	RSPI0	SPRI0	39	009Ch	Edge	○	○	○	×	×	IER04.IEN7	IPR039	DTCER039
○	○	○		SPTI0	40	00A0h	Edge	○	○	○	×	×	IER05.IEN0		DTCER040
○	○	○		SPII0	41	00A4h	Level	○	×	×	×	×	IER05.IEN1		—
○	○	○	RSPI1	SPRI1	42	00A8h	Edge	○	○	○	×	×	IER05.IEN2	IPR042	DTCER042
○	○	○		SPTI1	43	00ACh	Edge	○	○	○	×	×	IER05.IEN3		DTCER043
○	○	○		SPII1	44	00B0h	Level	○	×	×	×	×	IER05.IEN4		—
○	○	—	RSPI2	SPRI2	45	00B4h	Edge	○	○	○	×	×	IER05.IEN5	IPR045	DTCER045
○	○	—		SPTI2	46	00B8h	Edge	○	○	○	×	×	IER05.IEN6		DTCER046
○	○	—		SPII2	47	00BCh	Level	○	×	×	×	×	IER05.IEN7		—
○	○	○ *2	CAN0	RXF0	48	00C0h	Edge	○	×	×	×	×	IER06.IEN0	IPR048	—
○	○	○ *2		TXF0	49	00C4h	Edge	○	×	×	×	×	IER06.IEN1		—
○	○	○ *2		RXM0	50	00C8h	Edge	○	×	×	×	×	IER06.IEN2		—
○	○	○ *2		TXM0	51	00CCh	Edge	○	×	×	×	×	IER06.IEN3		—
○	○	○	CAN1	RXF1	52	00D0h	Edge	○	×	×	×	×	IER06.IEN4	IPR052	—
○	○	○		TXF1	53	00D4h	Edge	○	×	×	×	×	IER06.IEN5		—
○	○	○		RXM1	54	00D8h	Edge	○	×	×	×	×	IER06.IEN6		—
○	○	○		TXM1	55	00DCh	Edge	○	×	×	×	×	IER06.IEN7		—

Table 15.3 Interrupt Vector Table (3/7)

177/176-pin package	145/144-pin package	100-pin package	Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
○ *3	○ *3	—	CAN2	RXF2	56	00E0h	Edge	○	×	×	×	×	IER07.IEN0	IPR056	—
○ *3	○ *3	—		TXF2	57	00E4h	Edge	○	×	×	×	×	IER07.IEN1		—
○ *3	○ *3	—		RXM2	58	00E8h	Edge	○	×	×	×	×	IER07.IEN2		—
○ *3	○ *3	—		TXM2	59	00ECh	Edge	○	×	×	×	×	IER07.IEN3		—
—	—	—	—	Reserved	60	00F0h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	61	00F4h	—	×	×	×	×	×	—	—	—
○	○	○	RTC	COUNTUP	62	00F8h	Edge	○	×	×	×	×	IER07.IEN6	IPR062	—
—	—	—	—	Reserved	63	00FCh	—	×	×	×	×	×	—	—	—
○	○	○	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
—	—	—	—	Reserved	80	0140h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	81	0144h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	82	0148h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	83	014Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	84	0150h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	85	0154h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	86	0158h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	87	015Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	88	0160h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	89	0164h	—	×	×	×	×	×	—	—	—
○	○	○	USB	USBR0	90	0168h	Level	○	×	×	○	○	IER0B.IEN2	IPR090	—
○	—	—		USBR1	91	016Ch	Level	○	×	×	○	○	IER0B.IEN3	IPR091	—
○	○	○	RTC	ALM	92	0170h	Edge	○	×	×	○	○	IER0B.IEN4	IPR092	—
○	○	○		PRD	93	0174h	Edge	○	×	×	○	○	IER0B.IEN5	IPR093	—
—	—	—	—	Reserved	94	0178h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	95	017Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	96	0180h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	97	0184h	—	×	×	×	×	×	—	—	—
○	○	○	AD0	ADI0	98	0188h	Edge	○	○	○	×	×	IER0C.IEN2	IPR098	DTCER098
—	—	—	—	Reserved	99	018Ch	—	×	×	×	×	×	—	—	—

Table 15.3 Interrupt Vector Table (4/7)

177/176-pin package	145/144-pin package	100-pin package	Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
—	—	—	—	Reserved	100	0190h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	101	0194h	—	×	×	×	×	×	—	—	—
○	○	○	S12AD0	S12ADI0	102	0198h	Edge	○	○	○	×	×	IER0C.IEN6	IPR102	DTCER102
—	—	—	—	Reserved	103	019Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	104	01A0h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	105	01A4h	—	×	×	×	×	×	—	—	—
○	○	○	ICU*4	GROUP0	106	01A8h	Level	○	×	×	×	×	IER0D.IEN2	IPR106	—
○	○	○		GROUP1	107	01ACh	Level	○	×	×	×	×	IER0D.IEN3	IPR107	—
○	○	○		GROUP2	108	01B0h	Level	○	×	×	×	×	IER0D.IEN4	IPR108	—
○	○	○		GROUP3	109	01B4h	Level	○	×	×	×	×	IER0D.IEN5	IPR109	—
○	○	○		GROUP4	110	01B8h	Level	○	×	×	×	×	IER0D.IEN6	IPR110	—
○	○	—		GROUP5	111	01BCh	Level	○	×	×	×	×	IER0D.IEN7	IPR111	—
○	○	—		GROUP6	112	01C0h	Level	○	×	×	×	×	IER0E.IEN0	IPR112	—
—	—	—	—	Reserved	113	01C4h	—	×	×	×	×	×	—	—	—
○	○	○	ICU	GROUP12	114	01C8h	Level	○	×	×	×	×	IER0E.IEN2	IPR114	—
—	—	—	—	Reserved	115	01CCh	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	116	01D0h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	117	01D4h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	118	01D8h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	119	01DCh	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	120	01E0h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	121	01E4h	—	×	×	×	×	×	—	—	—
○	○	○	SCI12	SCIX0	122	01E8h	Level	○	×	×	×	×	IER0F.IEN2	IPR122	—
○	○	○		SCIX1	123	01ECh	Level	○	×	×	×	×	IER0F.IEN3		—
○	○	○		SCIX2	124	01F0h	Level	○	×	×	×	×	IER0F.IEN4		—
○	○	○		SCIX3	125	01F4h	Level	○	×	×	×	×	IER0F.IEN5		—
○	○	○	TPU0	TGI0A	126	01F8h	Edge	○	○	○	×	×	IER0F.IEN6	IPR126	DTCER126
○	○	○		TGI0B	127	01FCh	Edge	○	○	×	×	×	IER0F.IEN7		DTCER127
○	○	○		TGI0C	128	0200h	Edge	○	○	×	×	×	IER10.IEN0		DTCER128
○	○	○		TGI0D	129	0204h	Edge	○	○	×	×	×	IER10.IEN1		DTCER129
○	○	○	TPU1	TGI1A	130	0208h	Edge	○	○	○	×	×	IER10.IEN2	IPR130	DTCER130
○	○	○		TGI1B	131	020Ch	Edge	○	○	×	×	×	IER10.IEN3		DTCER131
○	○	○	TPU2	TGI2A	132	0210h	Edge	○	○	○	×	×	IER10.IEN4	IPR132	DTCER132
○	○	○		TGI2B	133	0214h	Edge	○	○	×	×	×	IER10.IEN5		DTCER133
○	○	○	TPU3	TGI3A	134	0218h	Edge	○	○	○	×	×	IER10.IEN6	IPR134	DTCER134
○	○	○		TGI3B	135	021Ch	Edge	○	○	×	×	×	IER10.IEN7		DTCER135
○	○	○		TGI3C	136	0220h	Edge	○	○	×	×	×	IER11.IEN0		DTCER136
○	○	○		TGI3D	137	0224h	Edge	○	○	×	×	×	IER11.IEN1		DTCER137
○	○	○	TPU4	TGI4A	138	0228h	Edge	○	○	○	×	×	IER11.IEN2	IPR138	DTCER138
○	○	○		TGI4B	139	022Ch	Edge	○	○	×	×	×	IER11.IEN3		DTCER139
○	○	○	TPU5	TGI5A	140	0230h	Edge	○	○	○	×	×	IER11.IEN4	IPR140	DTCER140
○	○	○		TGI5B	141	0234h	Edge	○	○	×	×	×	IER11.IEN5		DTCER141

Table 15.3 Interrupt Vector Table (5/7)

177/176-pin package	145/144-pin package	100-pin package	Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
○	○	○ *5	TPU6/MTU0	TGI6A(TPU6)/ TGIA0(MTU0)	142	0238h	Edge	○	○	○	×	×	IER11.IEN6	IPR142	DTCER142
○	○	○ *5		TGI6B(TPU6)/ TGIB0(MTU0)	143	023Ch	Edge	○	○	×	×	×	IER11.IEN7		DTCER143
○	○	○ *5		TGI6C(TPU6)/ TGIC0(MTU0)	144	0240h	Edge	○	○	×	×	×	IER12.IEN0		DTCER144
○	○	○ *5		TGI6D(TPU6)/ TGID0(MTU0)	145	0244h	Edge	○	○	×	×	×	IER12.IEN1		DTCER145
○	○	○ *5		TGIE0(MTU0)	146	0248h	Edge	○	×	×	×	×	IER12.IEN2	IPR146	—
○	○	○ *5		TGIF0(MTU0)	147	024Ch	Edge	○	×	×	×	×	IER12.IEN3		—
○	○	○ *5	TPU7/MTU1	TGI7A(TPU7)/ TGIA1(MTU1)	148	0250h	Edge	○	○	○	×	×	IER12.IEN4	IPR148	DTCER148
○	○	○ *5		TGI7B(TPU7)/ TGIB1(MTU1)	149	0254h	Edge	○	○	×	×	×	IER12.IEN5		DTCER149
○	○	○ *5	TPU8/MTU2	TGI8A(TPU8)/ TGIA2(MTU2)	150	0258h	Edge	○	○	○	×	×	IER12.IEN6	IPR150	DTCER150
○	○	○ *5		TGI8B(TPU8)/ TGIB2(MTU2)	151	025Ch	Edge	○	○	×	×	×	IER12.IEN7		DTCER151
○	○	○ *5	TPU9/MTU3	TGI9A(TPU9)/ TGIA3(MTU3)	152	0260h	Edge	○	○	○	×	×	IER13.IEN0	IPR152	DTCER152
○	○	○ *5		TGI9B(TPU9)/ TGIB3(MTU3)	153	0264h	Edge	○	○	×	×	×	IER13.IEN1		DTCER153
○	○	○ *5		TGI9C(TPU9)/ TGIC3(MTU3)	154	0268h	Edge	○	○	×	×	×	IER13.IEN2		DTCER154
○	○	○ *5		TGI9D(TPU9)/ TGID3(MTU3)	155	026Ch	Edge	○	○	×	×	×	IER13.IEN3		DTCER155
○	○	○ *5	TPU10/MTU4	TGI10A(TPU10)/ TGIA4(MTU4)	156	0270h	Edge	○	○	○	×	×	IER13.IEN4	IPR156	DTCER156
○	○	○ *5		TGI10B(TPU10)/ TGIB4(MTU4)	157	0274h	Edge	○	○	×	×	×	IER13.IEN5		DTCER157
○	○	○ *5		TGIC4(MTU4)	158	0278h	Edge	○	○	×	×	×	IER13.IEN6		DTCER158
○	○	○ *5		TGID4(MTU4)	159	027Ch	Edge	○	○	×	×	×	IER13.IEN7		DTCER159
○	○	○ *5		TCIV4(MTU4)	160	0280h	Edge	○	○	×	×	×	IER14.IEN0	IPR160	DTCER160
○	○	○ *5	TPU11/MTU5	TGIU5(MTU5)	161	0284h	Edge	○	○	×	×	×	IER14.IEN1	IPR161	DTCER161
○	○	○ *5		TGIV5(MTU5)	162	0288h	Edge	○	○	×	×	×	IER14.IEN2		DTCER162
○	○	○ *5		TGIW5(MTU5)	163	028Ch	Edge	○	○	×	×	×	IER14.IEN3		DTCER163
○	○	—		TGI11A(TPU11)	164	0290h	Edge	○	○	○	×	×	IER14.IEN4	IPR164	DTCER164
○	○	—		TGI11B(TPU11)	165	0294h	Edge	○	○	×	×	×	IER14.IEN5		DTCER165
○	○	○	POE	OEI1	166	0298h	Level	○	×	×	×	×	IER14.IEN6	IPR166	—
○	○	○		OEI2	167	029Ch	Level	○	×	×	×	×	IER14.IEN7		—
—	—	—	—	Reserved	168	02A0h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	169	02A4h	—	×	×	×	×	×	—	—	—
○	○	○	TMR0	CMIA0	170	02A8h	Edge	○	○	×	×	○*6	IER15.IEN2	IPR170	DTCER170
○	○	○		CMIB0	171	02ACh	Edge	○	○	×	×	○*6	IER15.IEN3		DTCER171
○	○	○		OVI0	172	02B0h	Edge	○	×	×	×	○*6	IER15.IEN4		—
○	○	○	TMR1	CMIA1	173	02B4h	Edge	○	○	×	×	○*6	IER15.IEN5	IPR173	DTCER173
○	○	○		CMIB1	174	02B8h	Edge	○	○	×	×	○*6	IER15.IEN6		DTCER174
○	○	○		OVI1	175	02BCh	Edge	○	×	×	×	○*6	IER15.IEN7		—

Table 15.3 Interrupt Vector Table (6/7)

177/176-pin package	145/144-pin package	100-pin package	Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
○	○	○	TMR2	CMIA2	176	02C0h	Edge	○	○	×	×	○*6	IER16.IEN0	IPR176	DTCER176
○	○	○		CMIB2	177	02C4h	Edge	○	○	×	×	○*6	IER16.IEN1		DTCER177
○	○	○		OVI2	178	02C8h	Edge	○	×	×	×	○*6	IER16.IEN2		—
○	○	○	TMR3	CMIA3	179	02CCh	Edge	○	○	×	×	○*6	IER16.IEN3	IPR179	DTCER179
○	○	○		CMIB3	180	02D0h	Edge	○	○	×	×	○*6	IER16.IEN4		DTCER180
○	○	○		OVI3	181	02D4h	Edge	○	×	×	×	○*6	IER16.IEN5		—
○	○	○	RIIC0	EEI0	182	02D8h	Level	○	×	×	×	×	IER16.IEN6	IPR182	—
○	○	○		RXI0	183	02DCh	Edge	○	○	○	×	×	IER16.IEN7	IPR183	DTCER183
○	○	○		TXI0	184	02E0h	Edge	○	○	○	×	×	IER17.IEN0	IPR184	DTCER184
○	○	○		TEI0	185	02E4h	Level	○	×	×	×	×	IER17.IEN1	IPR185	—
○	○	—	RIIC1	EEI1	186	02E8h	Level	○	×	×	×	×	IER17.IEN2	IPR186	—
○	○	—		RXI1	187	02ECh	Edge	○	○	○	×	×	IER17.IEN3	IPR187	DTCER187
○	○	—		TXI1	188	02F0h	Edge	○	○	○	×	×	IER17.IEN4	IPR188	DTCER188
○	○	—		TEI1	189	02F4h	Level	○	×	×	×	×	IER17.IEN5	IPR189	—
○	○	○	RIIC2	EEI2	190	02F8h	Level	○	×	×	×	×	IER17.IEN6	IPR190	—
○	○	○		RXI2	191	02FCh	Edge	○	○	○	×	×	IER17.IEN7	IPR191	DTCER191
○	○	○		TXI2	192	0300h	Edge	○	○	○	×	×	IER18.IEN0	IPR192	DTCER192
○	○	○		TEI2	193	0304h	Level	○	×	×	×	×	IER18.IEN1	IPR193	—
○	○	—	RIIC3	EEI3	194	0308h	Level	○	×	×	×	×	IER18.IEN2	IPR194	—
○	○	—		RXI3	195	030Ch	Edge	○	○	○	×	×	IER18.IEN3	IPR195	DTCER195
○	○	—		TXI3	196	0310h	Edge	○	○	○	×	×	IER18.IEN4	IPR196	DTCER196
○	○	—		TEI3	197	0314h	Level	○	×	×	×	×	IER18.IEN5	IPR197	—
○	○	○	DMAC	DMAC0I	198	0318h	Edge	○	○	×	×	×	IER18.IEN6	IPR198	DTCER198
○	○	○		DMAC1I	199	031Ch	Edge	○	○	×	×	×	IER18.IEN7	IPR199	DTCER199
○	○	○		DMAC2I	200	0320h	Edge	○	○	×	×	×	IER19.IEN0	IPR200	DTCER200
○	○	○		DMAC3I	201	0324h	Edge	○	○	×	×	×	IER19.IEN1	IPR201	DTCER201
○	○	—	EXDMAC	EXDMAC0I	202	0328h	Edge	○	○	×	×	×	IER19.IEN2	IPR202	DTCER202
○	○	—		EXDMAC1I	203	032Ch	Edge	○	○	×	×	×	IER19.IEN3	IPR203	DTCER203
—	—	—	—	Reserved	204	0330h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	205	0334h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	206	0338h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	207	033Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	208	0340h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	209	0344h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	210	0348h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	211	034Ch	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	212	0350h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	213	0354h	—	×	×	×	×	×	—	—	—
○	○	○	SCI0	RXI0	214	0358h	Edge	○	○	○	×	×	IER1A.IEN6	IPR214	DTCER214
○	○	○		TXI0	215	035Ch	Edge	○	○	○	×	×	IER1A.IEN7		DTCER215
○	○	○		TEI0	216	0360h	Level	○	×	×	×	×	IER1B.IEN0		—
○	○	○	SCI1	RXI1	217	0364h	Edge	○	○	○	×	×	IER1B.IEN1	IPR217	DTCER217
○	○	○		TXI1	218	0368h	Edge	○	○	○	×	×	IER1B.IEN2		DTCER218
○	○	○		TEI1	219	036Ch	Level	○	×	×	×	×	IER1B.IEN3		—

Table 15.3 Interrupt Vector Table (7/7)

177/176-pin package	145/144-pin package	100-pin package	Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
○	○	○	SCI2	RXI2	220	0370h	Edge	○	○	○	×	×	IER1B.IEN4	IPR220	DTCER220
○	○	○		TXI2	221	0374h	Edge	○	○	○	×	×	IER1B.IEN5		DTCER221
○	○	○		TEI2	222	0378h	Level	○	×	×	×	×	IER1B.IEN6		—
○	○	○	SCI3	RXI3	223	037Ch	Edge	○	○	○	×	×	IER1B.IEN7	IPR223	DTCER223
○	○	○		TXI3	224	0380h	Edge	○	○	○	×	×	IER1C.IEN0		DTCER224
○	○	○		TEI3	225	0384h	Level	○	×	×	×	×	IER1C.IEN1		—
○	○	—	SCI4	RXI4	226	0388h	Edge	○	○	○	×	×	IER1C.IEN2	IPR226	DTCER226
○	○	—		TXI4	227	038Ch	Edge	○	○	○	×	×	IER1C.IEN3		DTCER227
○	○	—		TEI4	228	0390h	Level	○	×	×	×	×	IER1C.IEN4		—
○	○	○	SCI5	RXI5	229	0394h	Edge	○	○	○	×	×	IER1C.IEN5	IPR229	DTCER229
○	○	○		TXI5	230	0398h	Edge	○	○	○	×	×	IER1C.IEN6		DTCER230
○	○	○		TEI5	231	039Ch	Level	○	×	×	×	×	IER1C.IEN7		—
○	○	○	SCI6	RXI6	232	03A0h	Edge	○	○	○	×	×	IER1D.IEN0	IPR232	DTCER232
○	○	○		TXI6	233	03A4h	Edge	○	○	○	×	×	IER1D.IEN1		DTCER233
○	○	○		TEI6	234	03A8h	Level	○	×	×	×	×	IER1D.IEN2		—
○	○	—	SCI7	RXI7	235	03ACh	Edge	○	○	○	×	×	IER1D.IEN3	IPR235	DTCER235
○	○	—		TXI7	236	03B0h	Edge	○	○	○	×	×	IER1D.IEN4		DTCER236
○	○	—		TEI7	237	03B4h	Level	○	×	×	×	×	IER1D.IEN5		—
○	○	○	SCI8	RXI8	238	03B8h	Edge	○	○	○	×	×	IER1D.IEN6	IPR238	DTCER238
○	○	○		TXI8	239	03BCh	Edge	○	○	○	×	×	IER1D.IEN7		DTCER239
○	○	○		TEI8	240	03C0h	Level	○	×	×	×	×	IER1E.IEN0		—
○	○	○	SCI9	RXI9	241	03C4h	Edge	○	○	○	×	×	IER1E.IEN1	IPR241	DTCER241
○	○	○		TXI9	242	03C8h	Edge	○	○	○	×	×	IER1E.IEN2		DTCER242
○	○	○		TEI9	243	03CCh	Level	○	×	×	×	×	IER1E.IEN3		—
○	○	—	SCI10	RXI10	244	03D0h	Edge	○	○	○	×	×	IER1E.IEN4	IPR244	DTCER244
○	○	—		TXI10	245	03D4h	Edge	○	○	○	×	×	IER1E.IEN5		DTCER245
○	○	—		TEI10	246	03D8h	Level	○	×	×	×	×	IER1E.IEN6		—
○	○	—	SCI11	RXI11	247	03DCh	Edge	○	○	○	×	×	IER1E.IEN7	IPR247	DTCER247
○	○	—		TXI11	248	03E0h	Edge	○	○	○	×	×	IER1F.IEN0		DTCER248
○	○	—		TEI11	249	03E4h	Level	○	×	×	×	×	IER1F.IEN1		—
○	○	○	SCI12	RXI12	250	03E8h	Edge	○	○	○	×	×	IER1F.IEN2	IPR250	DTCER250
○	○	○		TXI12	251	03ECh	Edge	○	○	○	×	×	IER1F.IEN3		DTCER251
○	○	○		TEI12	252	03F0h	Level	○	×	×	×	×	IER1F.IEN4		—
○	○	○	IEB	IEBINT	253	03F4h	Level	○	×	×	×	×	IER1F.IEN5	IPR253	—
—	—	—	—	Reserved	254	03F8h	—	×	×	×	×	×	—	—	—
—	—	—	—	Reserved	255	03FCh	—	×	×	×	×	×	—	—	—

Note 1. An interrupt source with a smaller vector number takes precedence.
Note 2. Devices with less than 512 Kbytes of ROM do not have a CAN0 module.
Note 3. Devices with less than 1 Mbyte of ROM do not have a CAN2 module.
Note 4. For the allocation of interrupt requests to the various interrupt groups, see Table 15.4, Group m Interrupt Requests.
Note 5. TPU5 to TPU11 are not included in 100-pin product.
Note 6. When the setting is set so that the ICLK is slower than the PCLKB, the TMR interrupt cannot be used for canceling all-module clock stop mode.
Note 7. An interrupt source with a smaller vector number takes precedence.

15.3.2 Fast Interrupt Vector Table

The address of the entry in the interrupt vector table that corresponds to the vector number of the fast interrupt is placed in the fast interrupt vector register (FINTV) of the CPU.

15.3.3 Non-maskable Interrupt Vector Table

The non-maskable interrupt vector table is at FFFF FFF8h.

15.4 Peripheral Module Interrupt Request Groups and Unit Selection

15.4.1 Interrupt Request Groups

A maximum of 32 interrupt signals from peripheral modules are divided into groups and the interrupt signals in each group can be handled collectively as a single interrupt request.
Groups 0 to 6 comprise edge detection interrupt requests and group 12 comprises level detection interrupt requests.
When an interrupt request is detected and GENm.ENj bit (m = group number and j = bit number) is 1, the GRPm.ISj flag is set to 1.
When a GENm.ENj flag in group m is 1, the IRn.IR flag (n = interrupt vector number) corresponding to the group is set to 1.

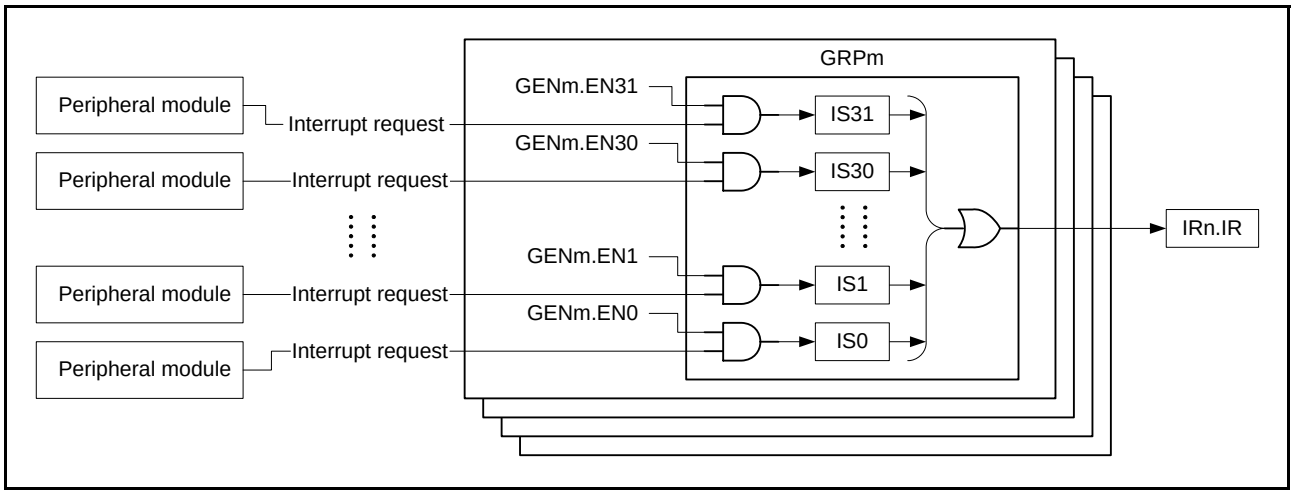


Figure 15.2 Interrupt Request Grouping

(1) Edge Detection Interrupt Requests

Groups 0 to 6 comprise edge detection interrupt requests.
When an interrupt request is detected and GENm.ENj bit (m = group number and j = bit number) is 1, the GRPm.ISj flag bit is set to 1. The interrupt request is not detected when GENm.ENj bit is 0.
The GRPm.ISj flag is cleared to 0 by writing 1 to the GCRm.CLRj bit.
The GRPm.ISj flag remains 1 if the GENm.ENj bit is cleared to 0 when the GRPm.ISj flag is 1.
For operation examples, see section 15.5.1.3, Edge Detection Group Interrupts and Interrupt Status Flags.

(2) Level Detection Interrupt Requests

Group 12 comprises level detection interrupt requests.
When an interrupt request is detected and GENm.ENj bit (m = group number and j = bit number) is 1, the GRPm.ISj flag bit is set to 1. When the interrupt request is detected as 0, the GRPm.ISj flag is also cleared to 0.
When the GENm.ENj bit is 0, the interrupt request is not detected and the GRPm.ISj flag is cleared to 0.
For operation examples, see section 15.5.1.4, Level Detection Group Interrupts and Interrupt Status Flags.

(3) List of Interrupt Requests for Each Group

Table 15.4 lists the interrupt requests by group.

Table 15.4 Group m Interrupt Requests (1/2)

Group	Interrupt Request Source Interrupt Request	Name	GENm.ENj Bit	GRPm.ISj Flag	GCRm.CLRj Flag	Vector No. (IRn.IR)
Group 0	CAN0	ERS0 (error interrupt)	GEN00.EN0	GRP00.IS0	GCR00.CLR0	106
	CAN1	ERS1 (error interrupt)	GEN00.EN1	GRP00.IS1	GCR00.CLR1	
	CAN2	ERS2 (error interrupt)	GEN00.EN2	GRP00.IS2	GCR00.CLR2	
Group 1	MTU0	TCIV0 (overflow)	GEN01.EN0	GRP01.IS0	GCR01.CLR0	107
	MTU1	TCIV1 (overflow)	GEN01.EN1	GRP01.IS1	GCR01.CLR1	
		TCIU1 (underflow)	GEN01.EN2	GRP01.IS2	GCR01.CLR2	
Group 2	MTU2	TCIV2 (overflow)	GEN02.EN0	GRP02.IS0	GCR02.CLR0	108
		TCIU2 (underflow)	GEN02.EN1	GRP02.IS1	GCR02.CLR1	
	MTU3	TCIV3 (overflow)	GEN02.EN2	GRP02.IS2	GCR02.CLR2	
Group 3	TPU0	TCI0V (overflow)	GEN03.EN0	GRP03.IS0	GCR03.CLR0	109
	TPU1	TCI1V (overflow)	GEN03.EN1	GRP03.IS1	GCR03.CLR1	
		TCI1U (underflow)	GEN03.EN2	GRP03.IS2	GCR03.CLR2	
	TPU5	TCI5V (overflow)	GEN03.EN3	IGRP03.S3	GCR03.CLR3	
		TCI5U (underflow)	GEN03.EN4	IGRP03.S4	GCR03.CLR4	
Group 4	TPU2	TCI2V (overflow)	GEN04.EN0	GRP04.IS0	GCR04.CLR0	110
		TCI2U (underflow)	GEN04.EN1	GRP04.IS1	GCR04.CLR1	
	TPU3	TCI3V (overflow)	GEN04.EN2	GRP04.IS2	GCR04.CLR2	
	TPU4	TCI4V (overflow)	GEN04.EN3	GRP04.IS3	GCR04.CLR3	
		TCI4U (underflow)	GEN04.EN4	GRP04.IS4	GCR04.CLR4	
Group 5	TPU6	TCI6V (overflow)	GEN05.EN0	GRP05.IS0	GCR05.CLR0	111
	TPU7	TCI7V (overflow)	GEN05.EN1	GRP05.IS1	GCR05.CLR1	
		TCI7U (underflow)	GEN05.EN2	GRP05.IS2	GCR05.CLR2	
	TPU11	TCI11V (overflow)	GEN05.EN3	GRP05.IS3	GCR05.CLR3	
		TCI11U (underflow)	GEN05.EN4	GRP05.IS4	GCR05.CLR4	
Group 6	TPU8	TCI8V (overflow)	GEN06.EN0	GRP06.IS0	GCR06.CLR0	112
		TCI8U (underflow)	GEN06.EN1	GRP06.IS1	GCR06.CLR1	
	TPU9	TCI9V (overflow)	GEN06.EN2	GRP06.IS2	GCR06.CLR2	
	TPU10	TCI10V (overflow)	GEN06.EN3	GRP06.IS3	GCR06.CLR3	
		TCI10U (underflow)	GEN06.EN4	GRP06.IS4	GCR06.CLR4	

Table 15.4 Group m Interrupt Requests (2/2)

Group	Interrupt Request Source Interrupt Request	Name	GENm.ENj Bit	GRPm.ISj Flag	GCRm.CLRj Flag	Vector No. (IRn.IR)
Group 12	SCI0	ERI0 (SCI0 reception error)	GEN12.EN0	GRP12.IS0	GCR12.CLR0	114
	SCI1	ERI1 (SCI1 reception error)	GEN12.EN1	GRP12.IS1	GCR12.CLR1	
	SCI2	ERI2 (SCI2 reception error)	GEN12.EN2	GRP12.IS2	GCR12.CLR2	
	SCI3	ERI3 (SCI3 reception error)	GEN12.EN3	GRP12.IS3	GCR12.CLR3	
	SCI4	ERI4 (SCI4 reception error)	GEN12.EN4	GRP12.IS4	GCR12.CLR4	
	SCI5	ERI5 (SCI5 reception error)	GEN12.EN5	GRP12.IS5	GCR12.CLR5	
	SCI6	ERI6 (SCI6 reception error)	GEN12.EN6	GRP12.IS6	GCR12.CLR6	
	SCI7	ERI7 (SCI7 reception error)	GEN12.EN7	GRP12.IS7	GCR12.CLR7	
	SCI8	ERI8 (SCI8 reception error)	GEN12.EN8	GRP12.IS8	GCR12.CLR8	
	SCI9	ERI9 (SCI9 reception error)	GEN12.EN9	GRP12.IS9	GCR12.CLR9	
	SCI10	ERI10 (SCI10 reception error)	GEN12.EN10	GRP12.IS10	GCR12.CLR10	
	SCI11	ERI11 (SCI11 reception error)	GEN12.EN11	GRP12.IS11	GCR12.CLR11	
	SCI12	ERI12 (SCI12 reception error)	GEN12.EN12	GRP12.IS12	GCR12.CLR12	
	RSPI0	SPEI0 (error interrupt)	GEN12.EN13	GRP12.IS13	GCR12.CLR13	
	RSPI1	SPEI1 (error interrupt)	GEN12.EN14	GRP12.IS14	GCR12.CLR14	
	RSPI2	SPEI2 (error interrupt)	GEN12.EN15	GRP12.IS15	GCR12.CLR15	

15.4.2 Unit Selection

The interrupts from two peripheral modules are handled as one interrupt request. One of each unit pair (MTUn and TPUn) is selected using the SEL.CNj bit (j = 0 to 5) and the interrupt signal from the selected unit is subjected to detection.

For operation examples, see section 15.5.1.5, Unit Selection and Interrupt Status Flags.

Table 15.5 lists the correspondence between the unit pairs and specific interrupt requests.

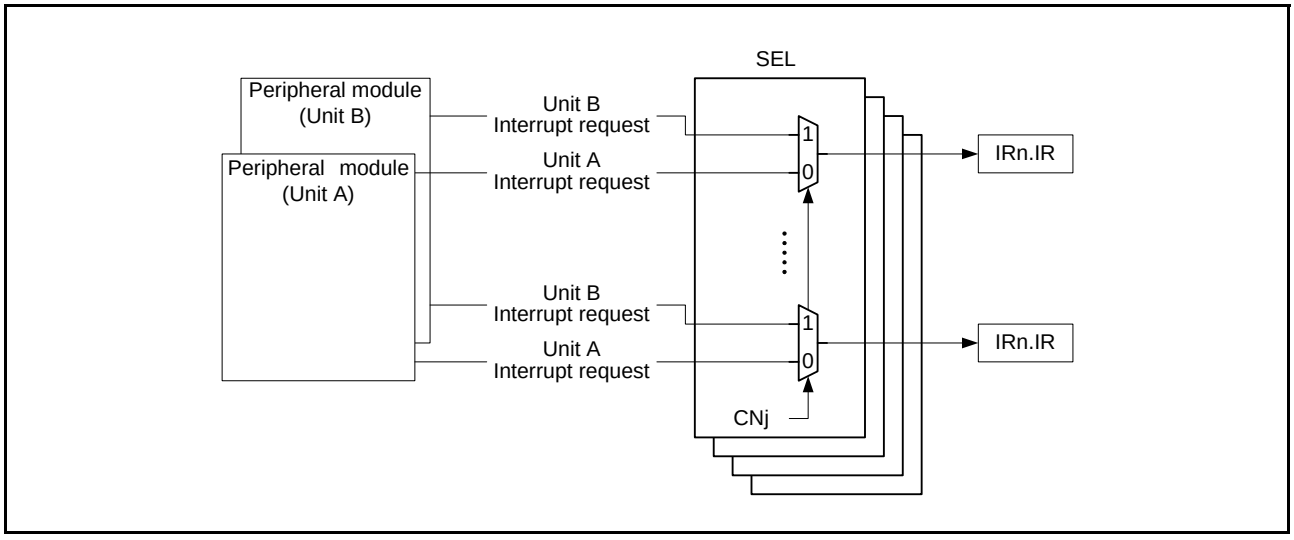


Figure 15.3 Unit Selection

Table 15.5 Unit n Interrupt Sources (n = 0 to 5)

Unit Pair No. (selected by SEL.CNj bit)	TPU		MTU2			
	Interrupt Request Source Interrupt Request	Name	Interrupt Request Source Interrupt Request	Name	Interrupt Type	Vector No. (IRn.IR)
Unit pair 0 (selected by SEL.CN0)	TPU6	TGI6A	MTU0	TGIA0	Input capture/compare match	142
	TPU6	TGI6B	MTU0	TGIB0	Input capture/compare match	143
	TPU6	TGI6C	MTU0	TGIC0	Input capture/compare match	144
	TPU6	TGI6D	MTU0	TGID0	Input capture/compare match	145
	—	—	MTU0	TGIE0	Compare match	146
	—	—	MTU0	TGIF0	Compare match	147
Unit pair 1 (selected by SEL.CN1)	TPU7	TGI7A	MTU1	TGIA1	Input capture/compare match	148
	TPU7	TGI7B	MTU1	TGIB1	Input capture/compare match	149
Unit pair 2 (selected by SEL.CN2)	TPU8	TGI8A	MTU2	TGIA2	Input capture/compare match	150
	TPU8	TGI8B	MTU2	TGIA2	Input capture/compare match	151
Unit pair 3 (selected by SEL.CN3)	TPU9	TGI9A	MTU3	TGIA3	Input capture/compare match	152
	TPU9	TGI9B	MTU3	TGIB3	Input capture/compare match	153
	TPU9	TGI9C	MTU3	TGIC3	Input capture/compare match	154
	TPU9	TGI9D	MTU3	TGID3	Input capture/compare match	155
Unit pair 4 (selected by SEL.CN4)	TPU10	TGI10A	MTU4	TGIA4	Input capture/compare match	156
	TPU10	TGI10B	MTU4	TGIB4	Input capture/compare match	157
	—	—	MTU4	TGIC4	Input capture/compare match	158
	—	—	MTU4	TGID4	Input capture/compare match	159
	—	—	MTU4	TCIV4	Overflow/underflow	160
Unit pair 5 (selected by SEL.CN5)	—	—	MTU5	TGIU5	Input capture/compare match	161
	—	—	MTU5	TGIV5	Input capture/compare match	162
	—	—	MTU5	TGIW5	Input capture/compare match	163
	TPU11	TGI11A	—	—	Input capture/compare match	164
	TPU11	TGI11B	—	—	Input capture/compare match	165

Note: • Interrupts to which only the TPUn or MTUn is allocated are valid when selected and masked when not selected.

15.5 Interrupt Operation

The interrupt controller performs the following processing.

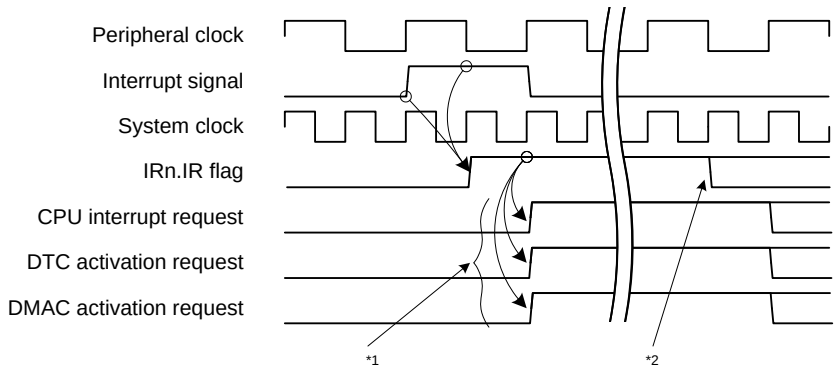
- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC activation, or DMAC activation)
- Determining priority

15.5.1 Detecting Interrupts

Interrupt requests are detected in either of two ways: the detection of edges of the interrupt signal or the detection of a level of the interrupt signal.
Edge detection or level detection is selected for the IRQi pins (i = 0 to 15) as external interrupt requests by the setting of the IRQMD[1:0] bits in IRQCRi.
For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source.
For the correspondence between interrupt sources and methods of detection, see Table 15.3, Interrupt Vector Table.
Grouped interrupt requests are classified into two groups; edge detection interrupt request group and level detection interrupt request group, and retained in the GRPm register (m = group number, 0 to 6 or 12). In both groups, the IRn.IR flag operates as the level interrupt. For details of the interrupt grouping function, see section 15.4, Peripheral Module Interrupt Request Groups and Unit Selection. For grouped interrupt requests, see Table 15.4, Group m Interrupt Requests.

15.5.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 15.4 shows the operation of the IR flag in IRn in the case of edge detection of an interrupt from a peripheral module or on an external pin.
The IR flag in IRn is set to 1 immediately after the transition of the interrupt signal due to generation of the interrupt. If the CPU is the request destination for the interrupt, the IR flag is automatically cleared to 0 on acceptance of the interrupt. If the DMAC or DTC is the request destination for the interrupt, the IRn.IR flag operation differs according to the DMAC/DTC transfer settings and transfer count. For details, see Table 15.6, Operation at DMAC/DTC Activation.



Note 1. One of the following requests is issued: CPU interrupt request, DTC activation request, and DMAC activation request. For details of the setting, see section 15.5.3, Selecting Interrupt Request Destinations.
Note 2. When the CPU interrupt request is specified, this flag is set to 0 on acceptance of a CPU interrupt. For the timing of 0 setting at the DTC activation or DMAC activation, see table 15.6 Operation at DMAC/DTC Activation.

Figure 15.4 IRn.IR Flag Operation for Edge Detection Interrupts

Figure 15.5 to Figure 15.8 show the interrupt signals of the interrupt controller. Note that the timings of the interrupts with interrupt vector numbers 64 to 95 are different from those of other interrupts. For the IRQ pin interrupts with interrupt vector numbers 64 to 79, “internal delay + 2 PCLK cycles” of delay is added after the IRQ pin input. For the interrupts with interrupt vector numbers 80 to 95, “2 PCLK cycles” of delay is added.

If an interrupt signal is generated every clock cycle, the subsequent interrupts cannot be detected; secure two or more clock cycles of the system clock or peripheral clock, whichever is slower, between issuance of continuous interrupt requests.

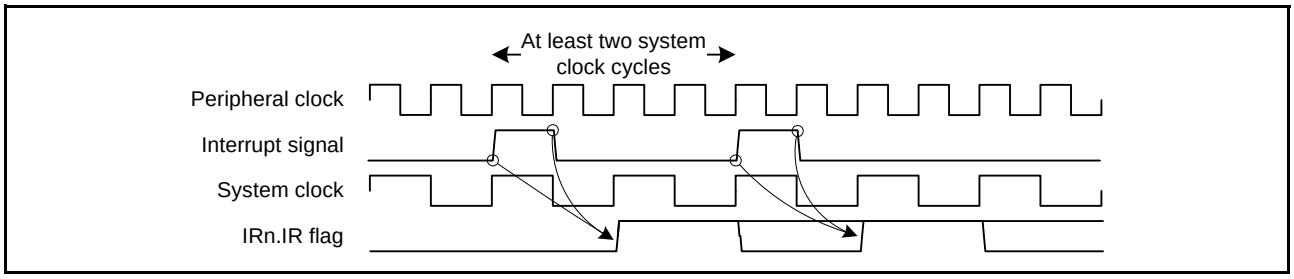


Figure 15.5 Interval Required between Issuance of Continuous Interrupt Requests (when the Frequency of System Clock is Slower than that of the Peripheral Clock)

While the IRn.IR flag is 1 after an interrupt request is generated, the interrupt request that is generated again will be ignored.*1

Figure 15.6 shows the timing for IRn.IR flag re-setting.

Note 1. When the transmission or reception interrupt of the SCI, RSPI, or RIIC is generated with the IRn.IR flag being 1, the interrupt request is retained. After the IRn.IR flag is cleared to 0, the IRn.IR flag is set to 1 again by the retained request. For details, see descriptions of the interrupts in section 34, Serial Communications Interface (SCIc, SCId), section 35, I^2C Bus Interface (RIIC), and section 37, Serial Peripheral Interface (RSPI).

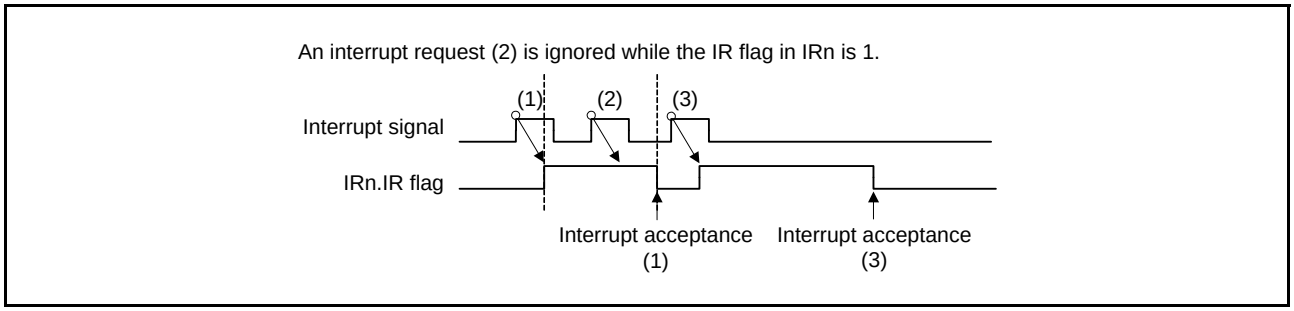


Figure 15.6 Timing for IRn.IR Flag Re-Setting

If an interrupt is disabled after the IRn.IR flag is set to 1 (output of the interrupt request is disabled by the interrupt enable bit of the relevant peripheral module), the IRn.IR flag is not affected but retains its state. Figure 15.7 shows operation when the interrupt is disabled.

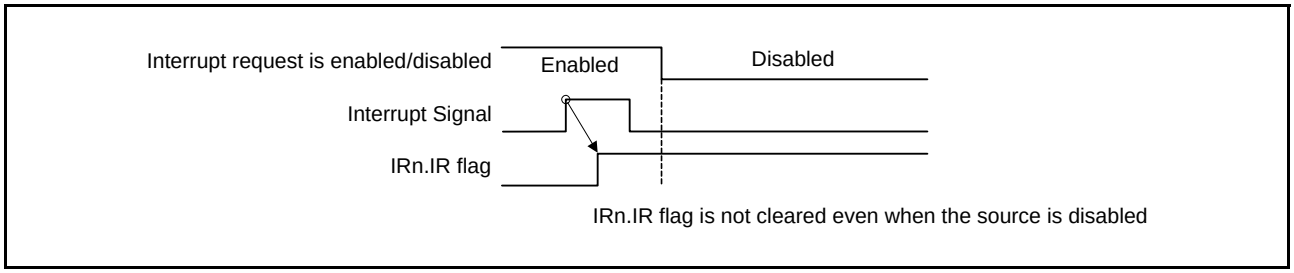


Figure 15.7 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

15.5.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 15.8 shows the operation of the interrupt status flag (IR flag) in IRn in the case of level detection of an interrupt from a peripheral module or an external pin.

The IR flag in IRn remains set to 1 as long as the interrupt signal is asserted. To clear the IRn.IR flag to 0, clear the interrupt request in the source generating the interrupt. Confirm that the interrupt request flag in the source generating the interrupt has been cleared to 0 and that the IRn.IR flag has been cleared to 0, and then complete the interrupt handling.

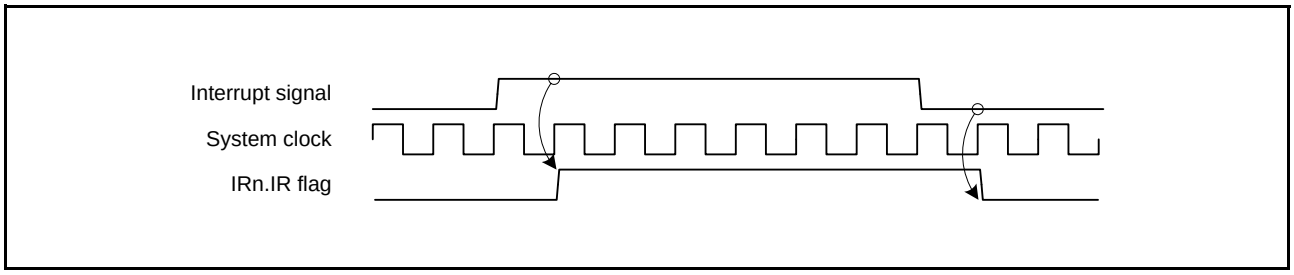


Figure 15.8 IRn.IR Flag Operation for Level Detection Interrupts

Figure 15.9 shows the procedure for handling level detection interrupts.

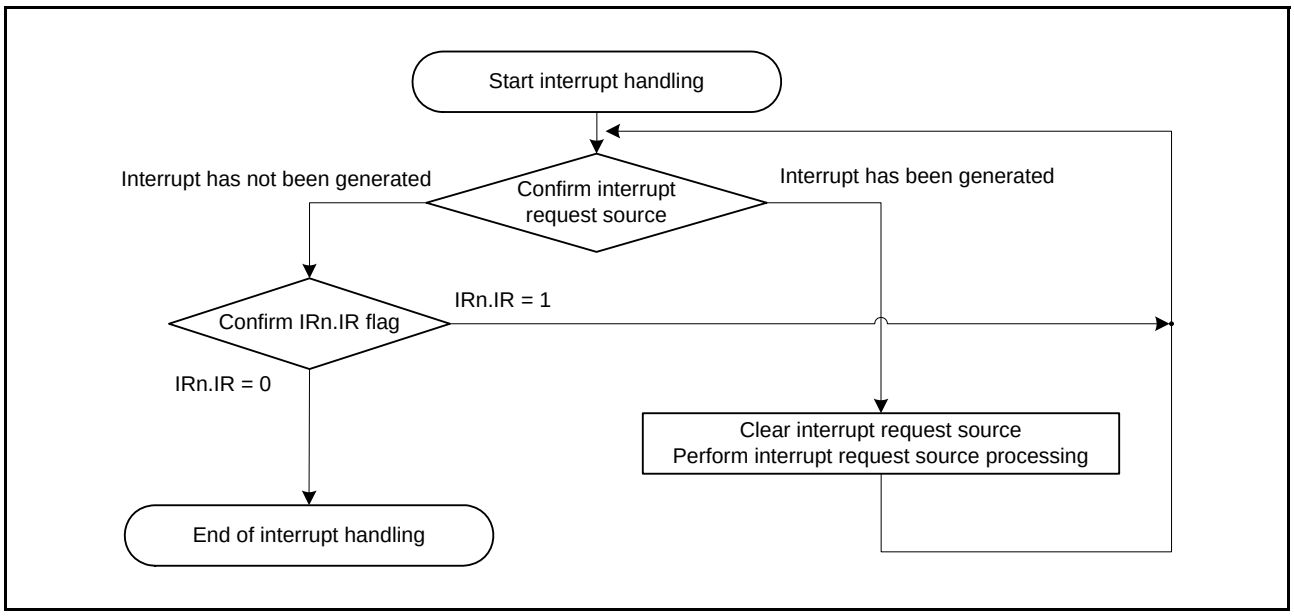


Figure 15.9 Procedure for Handling Level Detection Interrupts

15.5.1.3 Edge Detection Group Interrupts and Interrupt Status Flags

Edge detection interrupt requests are grouped into groups 0 to 6. The IRn.IR flag corresponding to the groups operates assuming the interrupt requests as level detection interrupts.

Figure 15.10 shows an operation example in which an edge detection interrupt request is generated and Figure 15.11 shows an operation example in which multiple edge detection interrupt requests allocated to a group are generated.

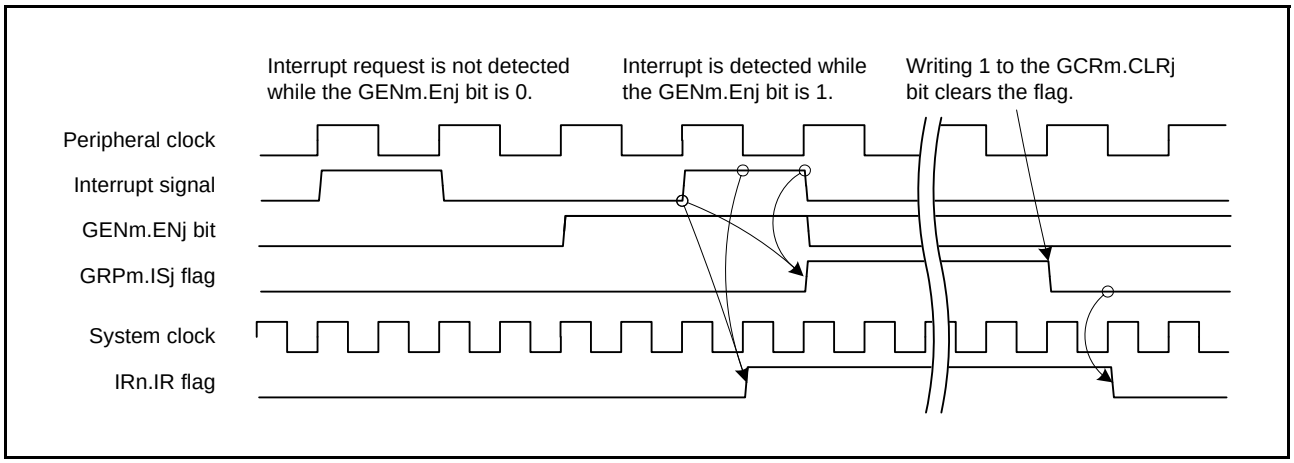


Figure 15.10 Operation Example in Which an Edge Detection Group Interrupt Request is Generated

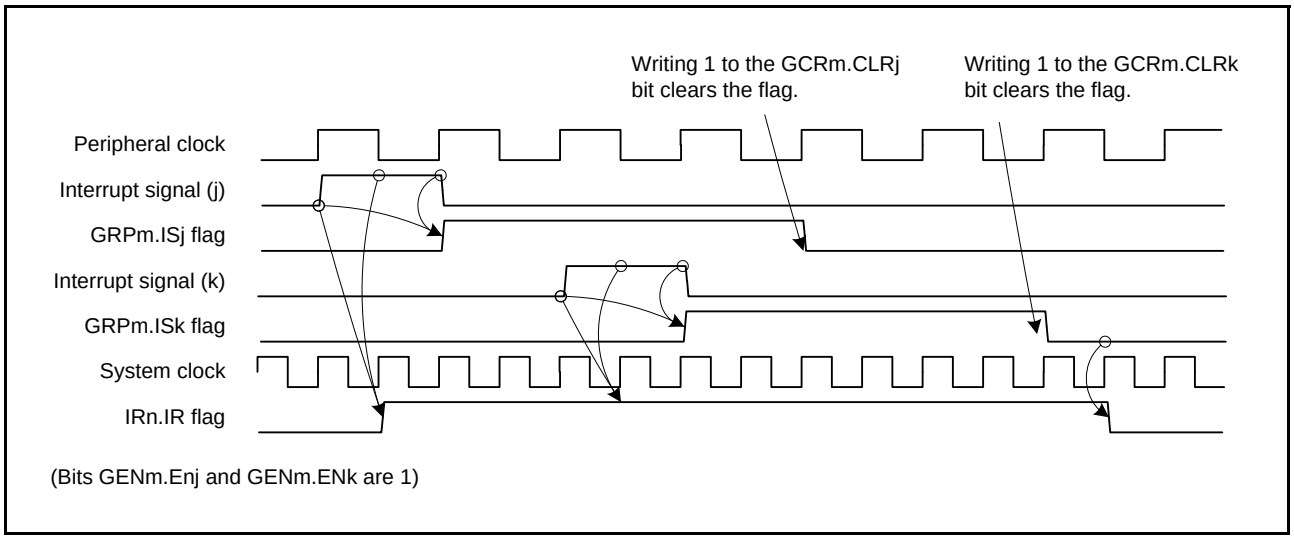


Figure 15.11 Operation Example in Which Multiple Edge Detection Interrupt Requests allocated to a Group are Generated

As for the interrupt handling of edge detection group 0, follow the procedure shown in Figure 15.12.
As for the interrupt handling of edge detection groups 1 to 6, follow the procedure shown in Figure 15.13.

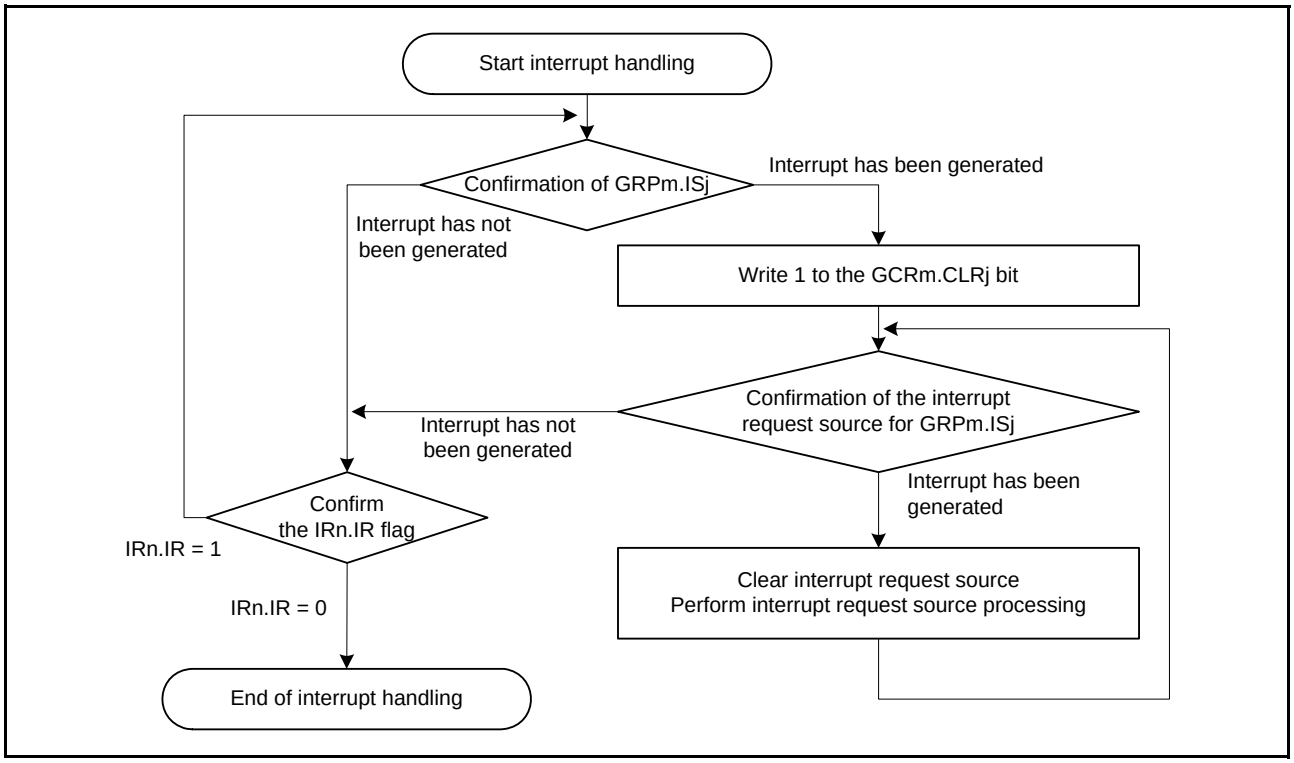


Figure 15.12 Procedure for Interrupt Handling of Edge Detection Group 0

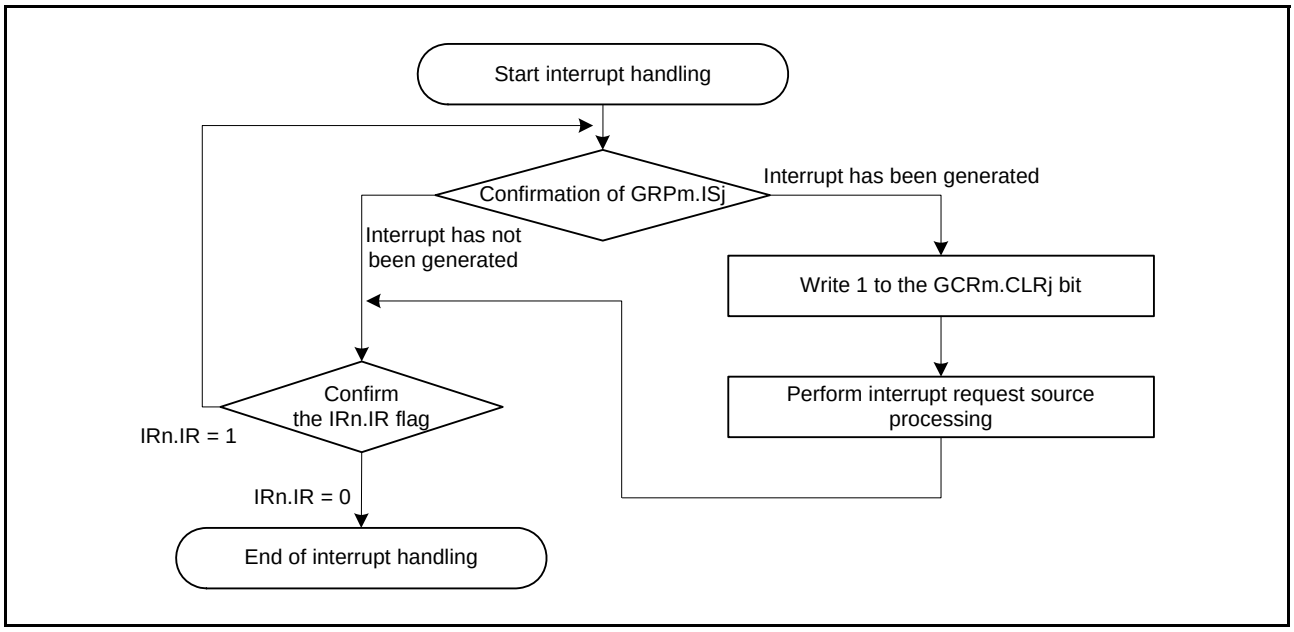


Figure 15.13 Procedure for Interrupt Handling of Edge Detection Groups 1 to 6

15.5.1.4 Level Detection Group Interrupts and Interrupt Status Flags

Level detection interrupt requests are grouped into group 12. The IRn.IR flag corresponding to the group operates assuming the interrupt requests as level detection interrupts.

Figure 15.14 shows an operation example in which a level detection interrupt request is generated and Figure 15.15 shows an operation example in which multiple level detection interrupt requests of a group are generated.

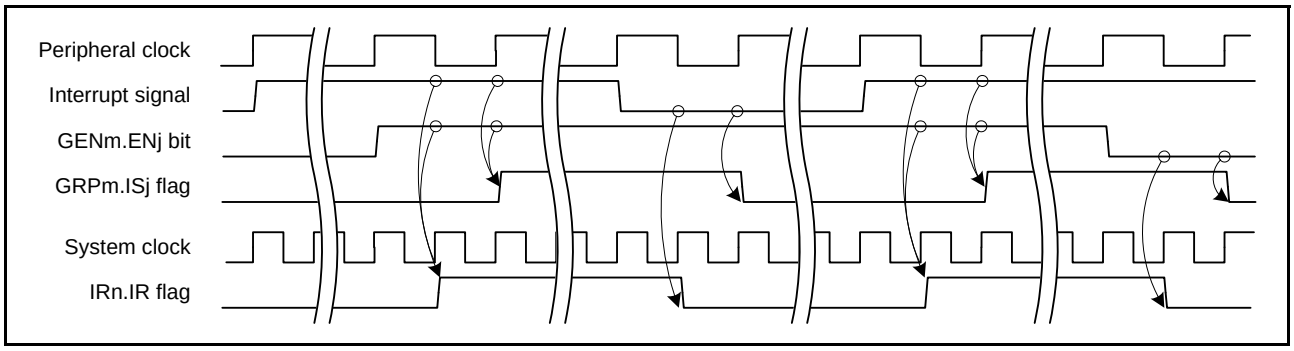


Figure 15.14 Operation Example in Which a Level Detection Group Interrupt Request is Generated

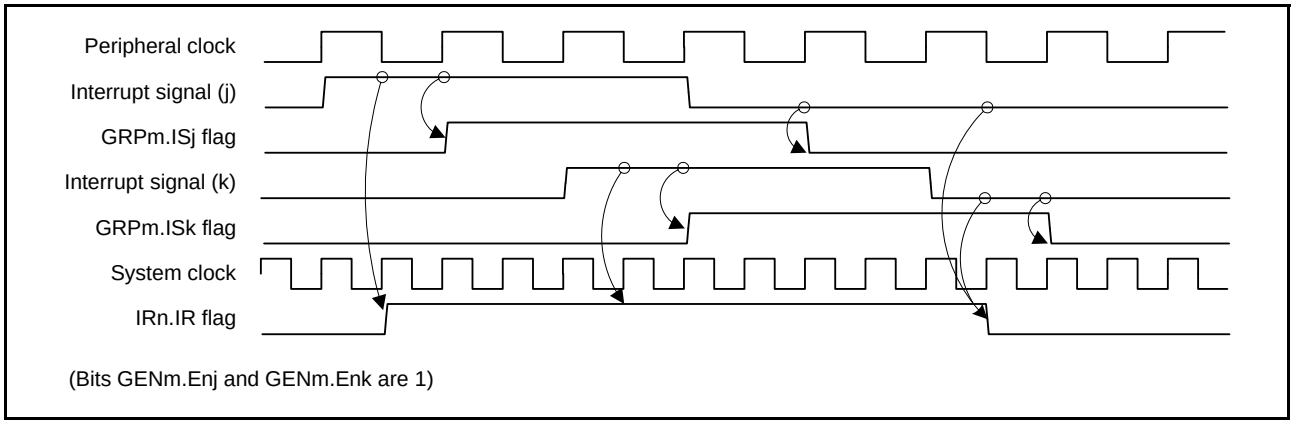


Figure 15.15 Operation Example in Which Multiple Level Detection Interrupt Requests of a Group are Generated

As for the interrupt handling of a level detection group, follow the procedure shown in Figure 15.16.

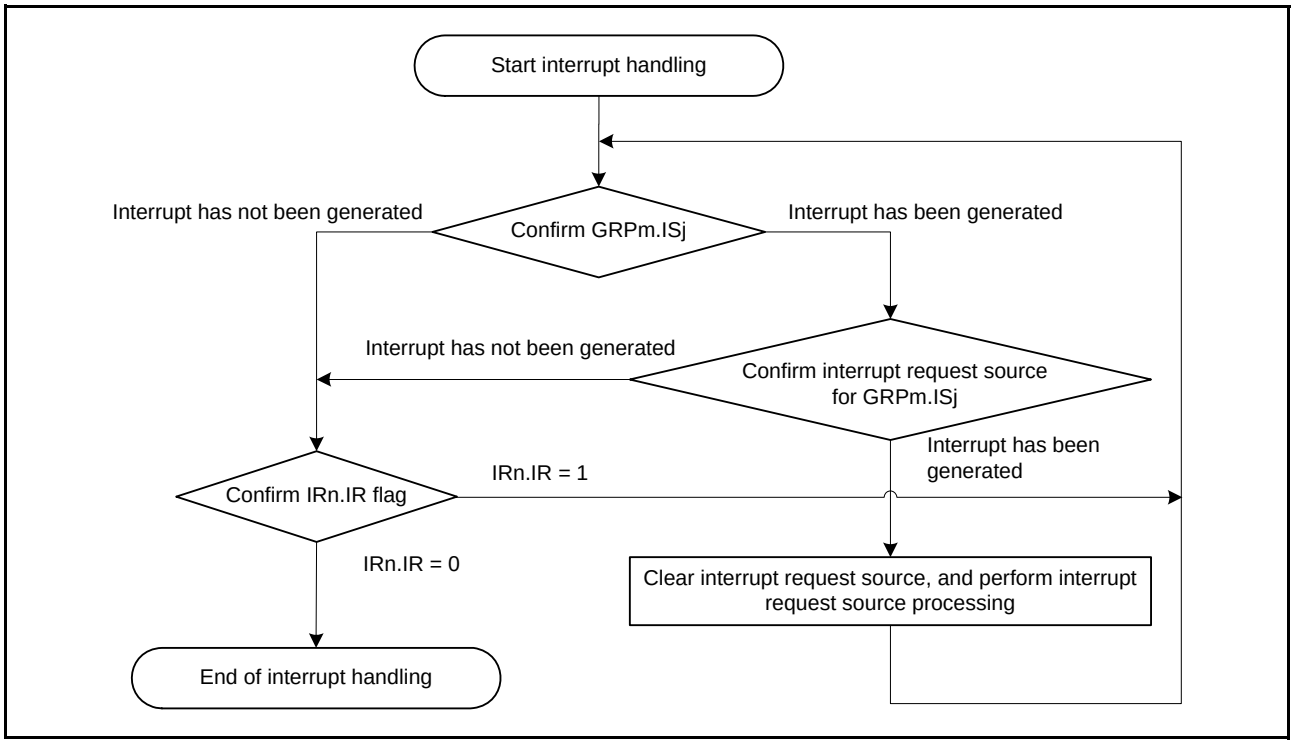


Figure 15.16 Procedure for Interrupt Handling of a Level Detection Group

15.5.1.5 Unit Selection and Interrupt Status Flags

Use the following procedure for switching units.

- Disable output of both MTUn (n = 0 to 5) and TPUn (n = 6 to 11) interrupts at the corresponding output sources of the interrupt requests, read the last written register, and confirm that writing is completed.
- Confirm that the IRn.IR flag is 0.
- Modify the SEL.CNj bit.
- Enable output of both MTUn (n = 0 to 5) and TPUn (n = 6 to 11) interrupts at the corresponding output sources of the interrupt requests.

Figure 15.17 shows a unit selection example.

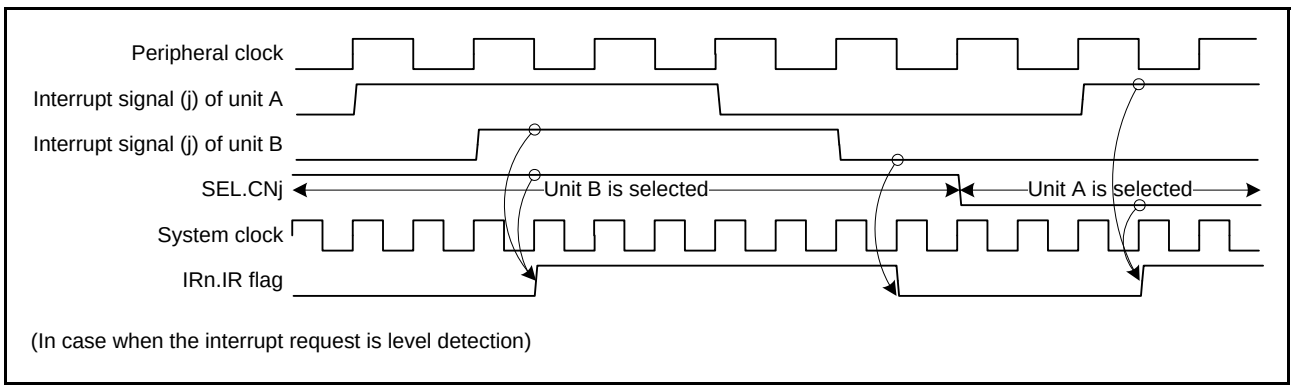


Figure 15.17 Unit Selection Example

15.5.2 Enabling and Disabling Interrupt Sources

Enabling requests from a given interrupt source requires the following settings.

1. In the case of interrupt requests from peripheral modules, setting the interrupt enable bit for the peripheral module to permit the output of interrupt requests from the source
2. For grouped interrupt requests, enabling the interrupts by the GENm.ENj bits
3. Enabling of the interrupt by the IERm.IENj bit

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag is set to 1. When a grouped interrupt request is generated, the corresponding GRPm.ISj flag is set to 1 and the IRn.IR flag corresponding to the group is set to 1.
Setting the IERm.IENj bit to enable an interrupt request allows the interrupt request for which the corresponding IRn.IR is 1 to be output to the interrupt request destination. Setting the IERm.IENj bit to disable an interrupt request suspends the output of the interrupt request for which the corresponding IRn.IR is 1.
The IRn.IR flag is not affected by the IERm.IENj bit.

Use the following procedure to disable interrupt requests.

1. Set the IERm.IENj bit to disable interrupt requests.
2. For grouped interrupt requests, set the GENm.ENj bit to disable.
3. Set the peripheral module interrupt output enable bit to disable the output. Read the last written register and confirm that writing is completed.
4. Check the IRn.IR flag, and clear the IRn.IR flag if necessary.*1
   For grouped interrupt requests, check the GRPm.ISj flag or clear the GRPm.ISj flag to 0.

Note 1. To disable the transmission or reception interrupt of the SCI, RSPI, or RIIC from the enabled state, clear the IRn.IR flag to 0 using the above procedure. For details, see descriptions of the interrupts in section 34, Serial Communications Interface (SCIc, SCId), section 35, I^2C Bus Interface (RIIC), and section 37, Serial Peripheral Interface (RSPI).

15.5.3 Selecting Interrupt Request Destinations

Possible settings for the request destination of each interrupt are fixed. That is, settings for request destination other than those indicated in Table 15.3, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a O in Table 15.3.
If the DMAC or DTC is selected as the destination for requests from an IRQ pin, be sure to set the IRQMD[1:0] bits in IRQCRi for that interrupt to select edge detection.
The following describes how to specify the destinations of interrupt requests.

(1) DMAC Activation

Make the following settings for each source while the IERm.IENj bit is 0.

1. Specify the vector number of the desired interrupt in the DMAC activation request select register (DMRSRm) for the required channel of the DMAC.*1
2. Set the activation source for the target DMAC channel (DMACm.DMTMD.DCTG[1:0]) to 01b (interrupt module detection).
3. Set the DMAC activation enable bit for the target DMAC channel (DMACm.DMCNT.DTE) to 1.

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DMAC operation enable bit (DMAST.DMST) to 1. The order of making settings for each interrupt and enabling the DMAC operation enable bit does not matter.
For the DMAC setting procedure, refer to section 17.3.7, Activating the DMAC in section 17, DMA Controller (DMACA).

(2) DTC Activation

Make the following settings for each source while the IERm.IENj bit is 0.

1. Set the DTC activation enable bit in the DTC activation enable register (DTCERn.DTCE) for the pertinent source to 1.*1

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DTC module activation bit (DTCST.DTCST) to 1. The order of making settings for each interrupt and enabling the DTC module activation bit does not matter.
For the DTC setting procedure, refer to section 19.5, DTC Setting Procedure, in section 19, Data Transfer Controller (DTCa).

Note 1. Do not set a DTC activation enable bit (DTCERn.DTCE) and a DMAC activation request select register (DMRSRm) to select the same source. Do not select the same source in more than one DMRSRm register.

(3) CPU Interrupt Request

If the interrupt request destination is neither the DMAC nor the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit to 1 while neither the DMAC activation settings nor the DTC activation settings described above are in place.

Table 15.6 shows operation when the DTC or the DMAC is the request destination.

Table 15.6 Operation at DMAC/DTC Activation

Interrupt Request Destination	DISEL	Remaining Number of Transfer Operations	Operation per Request	IR*1	Interrupt Request Destination after Transfer
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DMA transfer	Cleared at the start of DMAC transfer	DMAC
		= 0	DMA transfer*2	Cleared at the start of DMAC transfer*2	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DTCER.DTCE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer information	DTC
		= 0	DTC transfer → CPU interrupt *2	Cleared on interrupt acceptance by the CPU*2	The DTCER.DTCE bit is cleared and the CPU becomes the destination.

DISEL for the DMAC is set by the DMACm.DMCSL.DISEL bit; DISEL for the DTC is set by the DTC.MRB.DISEL bit.

Note 1. When the IRn.IR flag is 1, an interrupt request (DTC or DMAC activation request) that is generated again will be ignored.

Note 2. When the DISEL bit is 0, operation with the remaining number of transfer operations being 0 differs according to whether the source is for DTC or DMAC.

Note 3. For chain transfer, DTC transfer continues until the last chain transfer ends. Whether a CPU interrupt is generated at the end of chain transfer, the IRn.IR flag clear timing, and the interrupt request destination after transfer are determined by the state of DISEL and the remaining transfer count at the end of chain transfer. For the chain transfer, see Table 19.3, Chain Transfer Conditions in section 19, Data Transfer Controller (DTCa).

The request destination for an interrupt should be changed while the IERm.IENj bit is 0.

When a source is to be changed to an interrupt request or the DMA activation source is to be changed while a transfer is not complete (i.e. while the DMACm.DMCNT.DTE bit has not been cleared) after the settings described under (1) DMAC Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DMAC. If transfer is in progress, wait for its completion.
3. Make the settings described under (1) DMAC Activation.

When a source is to be changed to an interrupt request or the DTC transfer information is to be changed while a transfer is not complete (i.e. while the DTCERn.DTCE bit has not been cleared) after the settings described under (2) DTC Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DTC. If transfer is in progress, wait for its completion.
3. Make the settings described under (2) DTC Activation.

15.5.4 Determining Priority

Interrupt priority is determined for each interrupt request destination.
The priority for each interrupt request destination is determined as follows.

(1) Determining Priority when the CPU is the Request Destination of the Interrupt Signal

A source selected for the fast interrupt has the highest priority. After that, an interrupt source with a larger value of the interrupt priority level select bits (IPR[3:0]) in IPRn takes priority. If interrupts with the same priority level are generated by multiple sources, the source with the smallest vector number takes precedence.

(2) Determining Priority when the DTC is the Request Destination of the Interrupt Signal

The IPR[3:0] bits in IPRn have no effect. An interrupt source with a smaller vector number takes precedence.

(3) Determining Priority when the DMAC is the Request Destination of the Interrupt Signal

The IPR[3:0] bits in IPRn have no effect. Regarding the order of priority of DMAC channels, see section 17, DMA Controller (DMACA).

15.5.5 Fast Interrupt

The fast interrupt is a facility for faster interrupt processing by the CPU, so is only effective for an interrupt request being conveyed to the CPU. That is, the fast interrupt setting has no effect on interrupt requests for the DTC and DMAC.
The fast interrupt is set up by specifying the vector number of an interrupt source in the FVCT[7:0] bits in FIR and enabling the fast interrupt by setting the FIEN bit in FIR to 1. When the given source generates an interrupt, the interrupt is output to the CPU for handling as the fast interrupt.
The interrupt source selected for the fast interrupt has the highest priority regardless of the setting of the IPR[3:0] bits in IPRn.
For details on the fast interrupt, see section 14, Exception Handling.

15.5.6 Digital Filter

The digital filter function is provided for the external interrupt request IRQi pins (i = 0 to 15) and NMI pin interrupt.
The digital filter samples input signals at the filter sampling clock (PCLK) and removes the pulses of which length is less than three sampling cycles.
To use the digital filter for the IRQi pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the IRQFLTCn.FCLKSELi[1:0] bits (n = 0 or 1, i = 0 to 15) and set the IRQFLTEn.FLTENi bit to 1 (digital filter enabled).
To use the digital filter for the NMI pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the NMIFLTC.NFCLKSEL[1:0] bits and set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).
Figure 15.18 shows an example of digital filter operation.

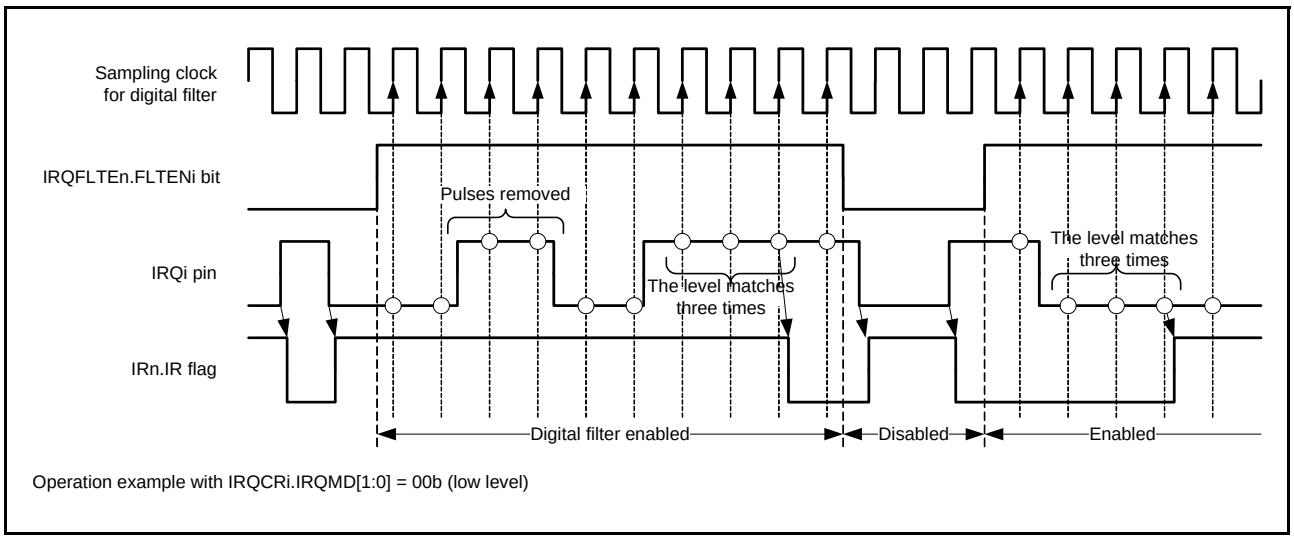


Figure 15.18 Digital Filter Operation Example

Before software standby mode is entered, set the IRQFLTEn.FLTENi and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after return from software standby mode, set the IRQFLTEn.FLTENi or NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

15.5.7 External Pin Interrupts

The procedure for using the signal on an external pin as an interrupt is as follows.

1. Clear the IERm.IENj bit to 0 (interrupt request disabled).
2. Clear the IRQFLTEn.FLTENi bit (n = 0 or 1 and i = 0 to 15) to 0 (digital filter disabled).
3. Set the digital filter sampling clock with the IRQFLTCn.FCLKSELi[1:0] bits.
4. Make or confirm the I/O port settings.
5. Set the method of detection for the interrupt in the IRQCRi.IRQMD[1:0] bits.
6. Clear the corresponding IRn.IR flag to 0 (if edge detection is in use).
7. Set the IRQFLTEn.FLTENi bit to 1 (digital filter enabled).
8. If the interrupt is to be used for DMAC activation, set the DMRSRm.DMRS[7:0] bits. If the interrupt is to be used for DTC activation, set the DTCERn.DTCE bit. The interrupt will be a CPU interrupt if neither of these settings is made.
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

15.6 Non-maskable Interrupt Operation

There are six types of non-maskable interrupt: the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error, IWDT underflow/refresh error, voltage-monitoring 1 interrupt, and voltage-monitoring 2 interrupt. Non-maskable interrupts are only usable as interrupts for the CPU; that is, they are not capable of DTC or DMAC activation. Non-maskable interrupts take precedence over all interrupts, including the fast interrupt.
Non-maskable interrupt requests are accepted regardless of the states of the I (interrupt-enable) bit and IPL[3:0] (processor interrupt priority level) bits in the PSW of the CPU. The current states of the non-maskable interrupts can be checked in the non-maskable interrupt status register (NMISR).
Confirm that all bits of the NMISR have returned to 0 from within the handler for the non-maskable interrupt.
Non-maskable interrupts are disabled by default. If a system is to use non-maskable interrupts, the following procedure must be followed at the beginning of program processing.

Non-maskable interrupt usage procedure:

1. Set the stack pointer (SP).
2. To use the NMI pin, clear the NMIFLTE.NFLTEN bit to 0 (digital filter disabled).
3. To use the NMI pin, set the digital filter sampling clock with the NMIFLTC.NFCLKSEL[1:0] bits.
4. To use the NMI pin, set the NMI pin detection sense with the NMICR.NMIMD bit.
5. To use the NMI pin, write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
6. To use the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).
7. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. The NMI interrupt cannot be disabled. It can be disabled only by a reset.
For the flow of non-maskable interrupt processing, see section 14, Exception Handling.

Writing 1 to the NMICLR.NMICLR bit clears the NMI status flag (NMISR.NMIST) to 0.
Writing 1 to the NMICLR.OSTCLR bit clears the oscillation stop detection interrupt status flag (NMISR.OSTST) to 0.
Writing 1 to the NMICLR.WDTCLR bit clears the WDT underflow/refresh error status flag (NMISR.WDTST) to 0.
Writing 1 to the NMICLR.IWDTCLR bit clears the IWDT underflow/refresh error status flag (NMISR.IWDTST) to 0.
Writing 1 to the NMICLR.LVD1CLR bit clears the voltage-monitoring 1 interrupt status flag (NMISR.LVD1ST) to 0.
Writing 1 to the NMICLR.LVD2CLR bit clears the voltage-monitoring 2 interrupt status flag (NMISR.LVD2ST) to 0.

15.7 Return from Power-Down States

The interrupt sources that can be used to return operation from sleep mode, all-module clock stop mode, or software standby mode are listed in Table 15.3, Interrupt Vector Table.
For details, refer to section 11, Low Power Consumption. The following describes how to use an interrupt to return operation from each power-down mode.

15.7.1 Return from Sleep Mode

If the interrupt controller is to return operation from sleep mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts

1. Select the CPU as the interrupt request destination.
2. Use the IENj bit in IERm to enable the given interrupt request.
3. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.
4. Use the ENj bit in GENm to enable the given grouped interrupt requests.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

15.7.2 Return from All-Module Clock Stop Mode

If the interrupt controller is to return operation from all-module clock stop mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts

1. Select the interrupt source that enables the return from the all-module clock stop mode.
2. Select the CPU as the interrupt request destination.
3. Use the IENj bit in IERm to enable the given interrupt request.
4. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

15.7.3 Return from Software Standby Mode

The interrupt controller can return operation from a non-maskable interrupt or an interrupt that enables the return from the software standby mode.
The conditions for the return are listed below.

- Interrupts

1. Select the interrupt source that enables the return from the software standby mode.
2. Select the CPU as the interrupt request destination.
3. Use the IENj bit in IERm to enable the given interrupt request.
4. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.
   (For the interrupt source specified as a fast interrupt, as well as setting the fast interrupt set register (FIR), the interrupt priority level (IPRn) should be set above the level set by IPL in the PSW of the CPU.)

Interrupt requests through the IRQ pins that do not satisfy the above conditions are not detected while the clock is stopped in software standby mode.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.
Use the following procedure to make a transition to/from software standby mode.

1. Before software standby mode is entered, disable the digital filter for the interrupt source as a return target (IRQFLTEn.FLTENi = 0, NMIFLTE.NFLTEN = 0).
2. To use the digital filter again after return from software standby mode, enable the digital filter (IRQFLTEn.FLTENi = 1, NMIFLTE.NFLTEN = 1).

15.8 Usage Notes

15.8.1 Point for Caution Regarding WAIT Instructions While the Non-Maskable Interrupt Is in Use

Confirm that all status flags in register NMISR are 0 whenever a WAIT instruction is to be issued.

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 (on-chip RAM, on-chip ROM) • Operates in synchronization with the system clock (ICLK)
	Operand bus	• Connected to the CPU (for operands) • Connected to on-chip memory (on-chip RAM, on-chip ROM) • Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	• Connected to on-chip RAM
	Memory bus 2	• Connected to on-chip ROM
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 (on-chip RAM, on-chip ROM) • Operates in synchronization with the system clock (ICLK)
Internal peripheral bus External 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 (USB) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 4	• Connected to peripheral modules (EDMAC and ETHERC) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 5	Reserved bus
	Internal peripheral bus 6	• Connected to on-chip ROM (P/E) and E2 DataFlash 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): 50 MHz (max.) The CSC (CS area controller) and the EXDMAC operate in synchronization with the BCLK.
SDCLK (SDRAM clock): 50 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.

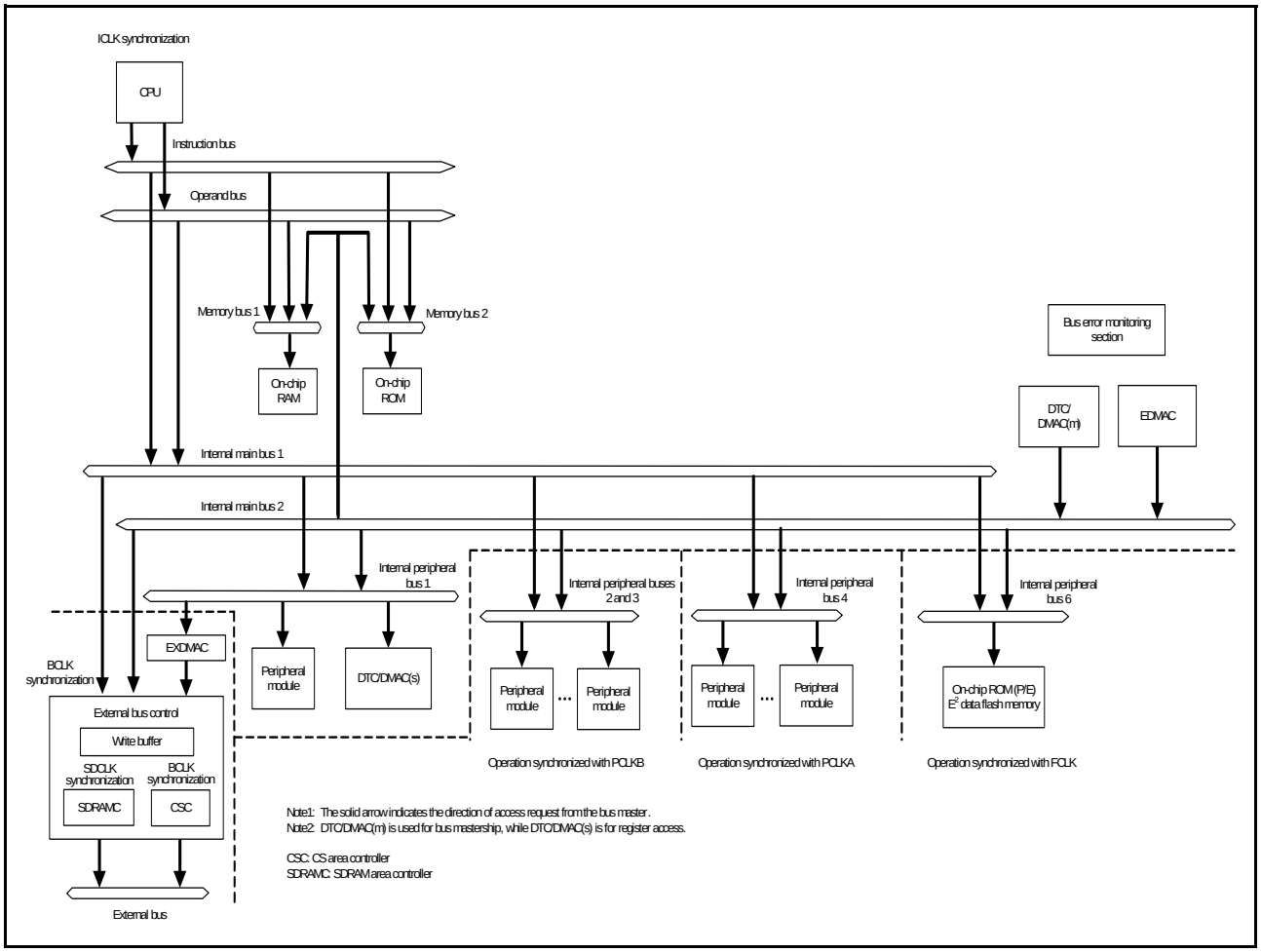


Figure 16.1 Bus Configuration

Table 16.2 Addresses Assigned for Each Bus

	Bus		Area	
	On-Chip ROM Mode		On-Chip ROM Mode	
Address	Enabled	Disabled	Enabled	Disabled
0000 0000h to 0001 FFFFh	Memory bus 1		On-chip RAM	
0002 0000h to 0007 FFFFh			Reserved area	
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	E2 DataFlash memory, FCU-RAM, and on-chip ROM (dedicated area for programming)	Reserved area
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	On-chip ROM (dedicated area for reading)	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. Connection of the instruction and operand buses to on-chip RAM and on-chip ROM 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 on-chip ROM 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, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to on-chip ROM and on-chip RAM or to on-chip ROM and external space is possible.

16.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. On-chip RAM is connected to memory bus 1 and on-chip ROM is connected to memory bus 2. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1 and 2.

The priority order of CPU bus and internal main bus 2 can be set using the memory bus 1 (on-chip RAM) priority control bits (BPRA[1:0]) and memory bus 2 (on-chip ROM) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. 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 activation request issues the bus mastership request. The priority order of activation 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	Bus Master
High ↑	EDMAC
	DMACA
	DTC
Low	CPU

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	USB
Internal peripheral bus 4	EDMAC and ETHERC
Internal peripheral bus 5	Bus with reserved function
Internal peripheral bus 6	On-chip ROM (P/E) /E2 DataFlash 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.
The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 16.2).

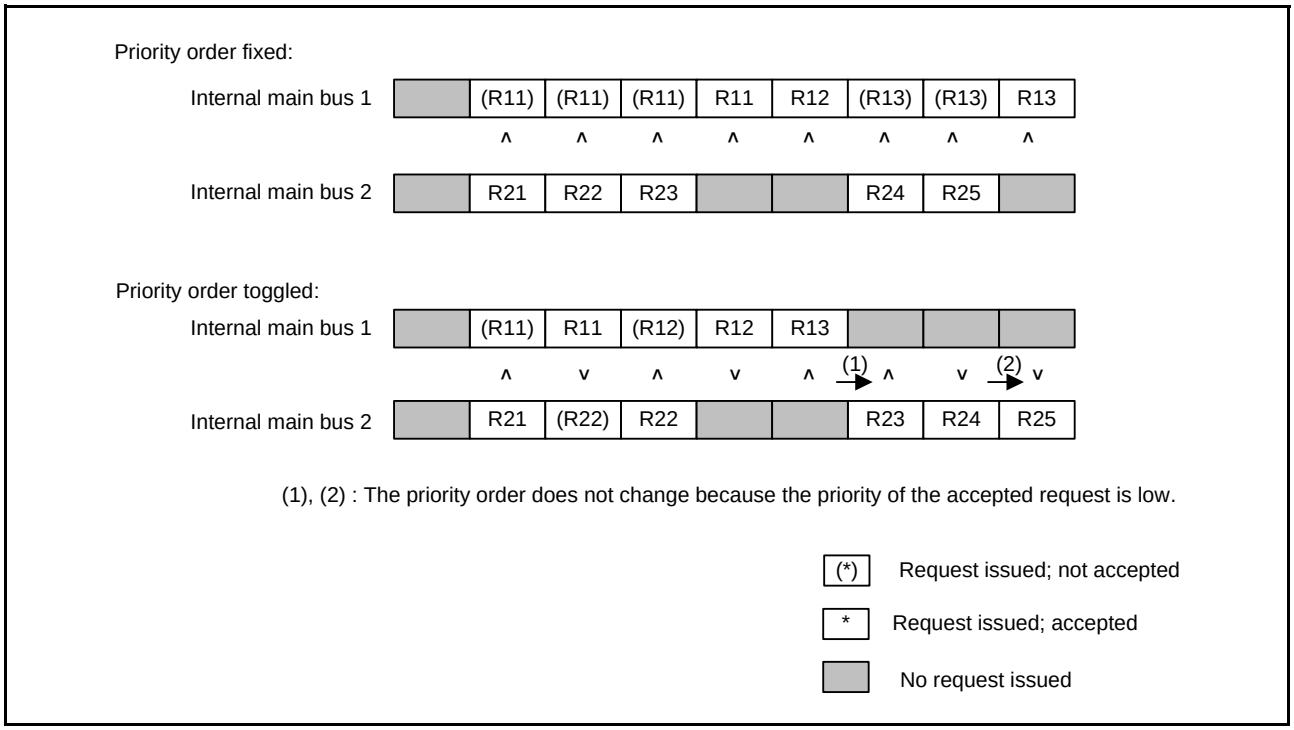


Figure 16.2 Priority Order between Internal Peripheral Bus Accesses

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 read 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).

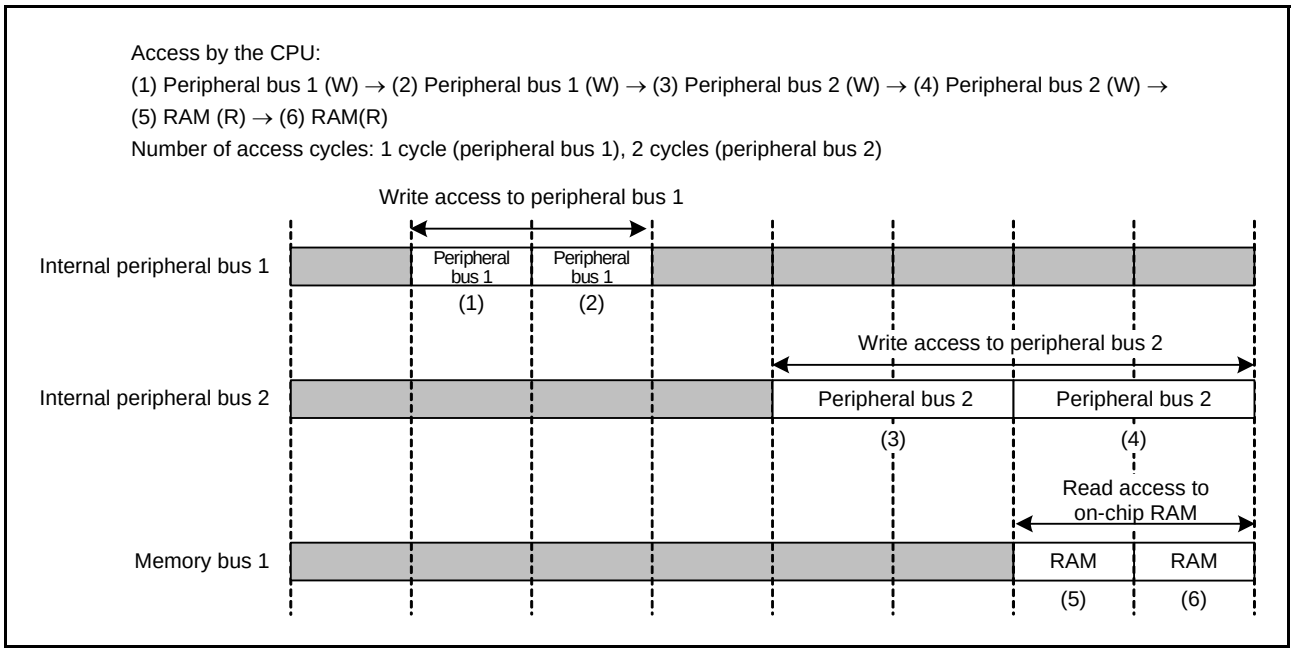


Figure 16.3 Write Buffer Function

16.2.5 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).

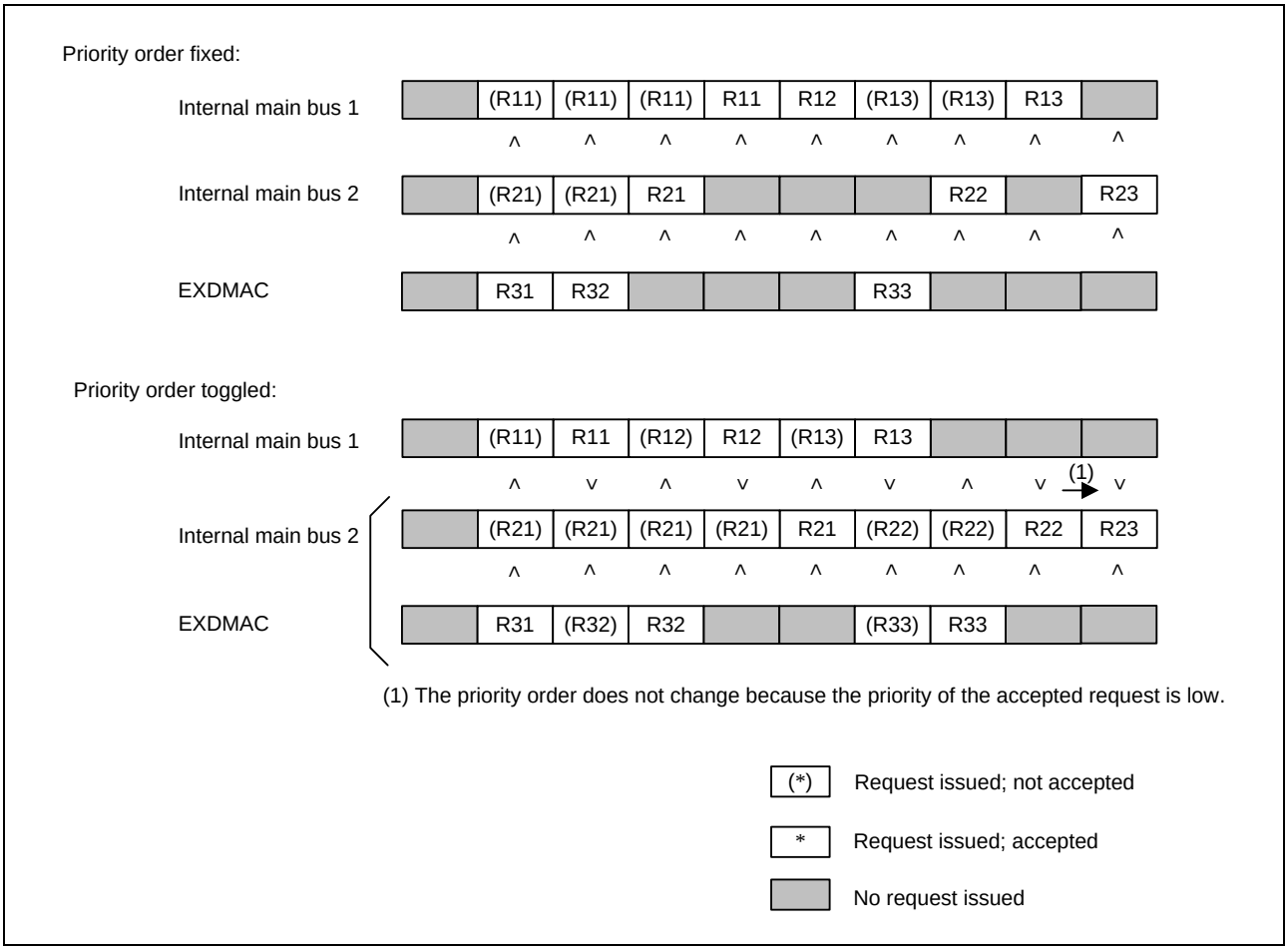


Figure 16.4 Priority Order of Internal Peripheral Bus Accesses

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.

Table 16.5 Specifications of the External Bus

Item	Description
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 (WR#, WR0# 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

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)
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

Table 16.6 Pin Configuration of the External Bus

Pin Name	I/O	Description
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 20, 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.

16.2.6 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 on-chip ROM and an operand from on-chip 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 on-chip ROM and on-chip RAM, respectively. Furthermore, the DMACA simultaneously employs internal main bus 2 for access to a peripheral bus or the external bus during access to on-chip RAM and ROM by the CPU.

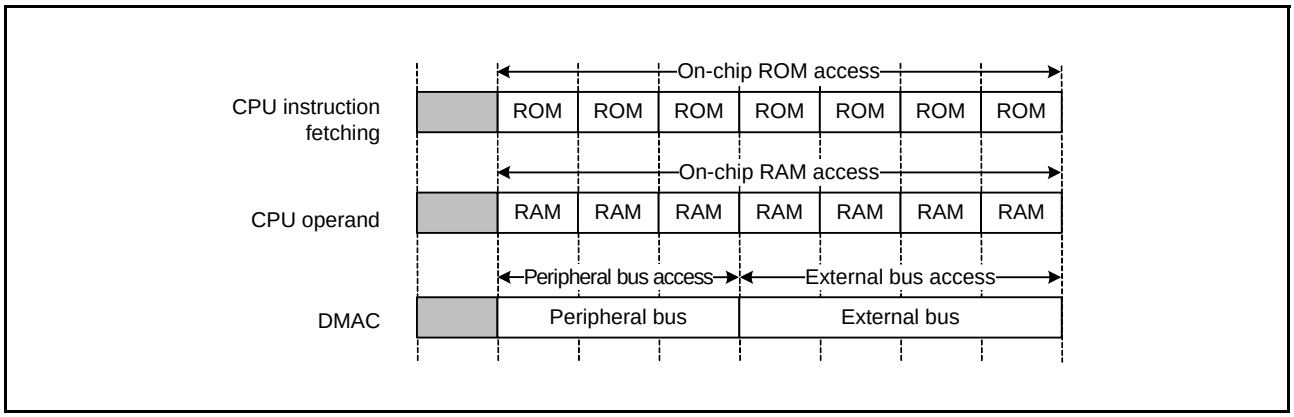


Figure 16.5 Example of Parallel Operations

16.2.7 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 selection register 0 (PFCSS0), CS output pin selection 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).

16.2.8 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.

(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	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*1
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*2
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

Note 1. The initial value of the EXENB bit in CS0CR is 1 and that in CSnCR (n = 1 to 7) is 0.
Note 2. The initial value of the BSIZE[1:0] bits in CS0CR is 10b.

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: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable	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 0) (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 0) (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

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 read accesses. Page read 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 is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is 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 is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is 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 is inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles is inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles is inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles is inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles is inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles is inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles is inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles is inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles is inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles is inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles is inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles is inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles is inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles is inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles is inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles is inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles is inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles is inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles is inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles is inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles is inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles is inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles is inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles is inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles is inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles is inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles is inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles is inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles is inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles is 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 is inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles is inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles is inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles is inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles is inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles is inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles is inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles is inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles is inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles is inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles is inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles is inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles is inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles is inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles is inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles is inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles is inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles is inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles is inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles is inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles is inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles is inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles is inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles is inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles is inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles is inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles is inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles is inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles is inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles is 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 is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is 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 cycle is inserted. 0 1 1: Wait with a length of 3 clock cycle is inserted. 1 0 0: Wait with a length of 4 clock cycle is inserted. 1 0 1: Wait with a length of 5 clock cycle is inserted. 1 1 0: Wait with a length of 6 clock cycle is inserted. 1 1 1: Wait with a length of 7 clock cycle is 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 is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b11	—	Reserved	These bits are 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 is inserted. 1 1: Wait with a length of 3 clock cycles is 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 cycle is inserted. 0 1 1: Wait with a length of 3 clock cycle is inserted. 1 0 0: Wait with a length of 4 clock cycle is inserted. 1 0 1: Wait with a length of 5 clock cycle is inserted. 1 1 0: Wait with a length of 6 clock cycle is inserted. 1 1 1: Wait with a length of 7 clock cycle is 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 cycle is inserted. 0 1 1: Wait with a length of 3 clock cycle is inserted. 1 0 0: Wait with a length of 4 clock cycle is inserted. 1 0 1: Wait with a length of 5 clock cycle is inserted. 1 1 0: Wait with a length of 6 clock cycle is inserted. 1 1 1: Wait with a length of 7 clock cycle is 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 cycle is inserted. 0 1 1: Wait with a length of 3 clock cycle is inserted. 1 0 0: Wait with a length of 4 clock cycle is inserted. 1 0 1: Wait with a length of 5 clock cycle is inserted. 1 1 0: Wait with a length of 6 clock cycle is inserted. 1 1 1: Wait with a length of 7 clock cycle is 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 cycle is inserted. 0 1 1: Wait with a length of 3 clock cycle is inserted. 1 0 0: Wait with a length of 4 clock cycle is inserted. 1 0 1: Wait with a length of 5 clock cycle is inserted. 1 1 0: Wait with a length of 6 clock cycle is inserted. 1 1 1: Wait with a length of 7 clock cycle is 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[1:0] 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[4:0] 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[4:0] 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: 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 always 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 always 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 cleared 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: 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 always read as 0. The write value should be 0.	R/W

Do not attempt to write the SDMOD register while the external bus is being accessed.

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: 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 always 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: 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 always 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 cleared 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: 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: 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 always read as 0. The write value should be 0.	R/W

RFEN Bit (Auto-Refresh Operation Enable)

Clearing this bit to 0 while auto-refreshing is enabled causes RFEN to be cleared 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 cleared 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: 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 always read as 0. The write value should be 0.	R/W

Do not attempt to write the SDICR register while the external bus is being accessed.

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: 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 always read as 0. The write value should be 0.	R/W

Do not attempt to write the SDIR register while the external bus is being accessed.

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: 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 always 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: 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 always 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 always 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 always 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.

Do not attempt to write the SDTR register while the external bus is being accessed.

CL[2:0] Bits (SDRAM 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: 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 always 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.
Do not attempt to write the SDMOD register while the external bus is being accessed.

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: 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 always 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 always 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 cleared to 0), bus access can cause the bus to freeze.
Note 2. Do not clear 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: Setting prohibited 0 1 0: Setting prohibited 0 1 1: DTC/DMAC 1 0 0: Setting prohibited 1 0 1: Setting prohibited 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 (On-Chip RAM) 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 (On-Chip ROM) 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 stopped. When they are written to more than one time, the operation is not guaranteed.

BPRA[1:0] Bits (Memory Bus 1 (On-Chip RAM) Priority Control)

These bits specify the priority order for memory bus 1 (on-chip RAM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPRO[1:0] Bits (Memory Bus 2 (On-Chip ROM) Priority Control)

These bits specify the priority order for memory bus 2 (on-chip ROM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

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).

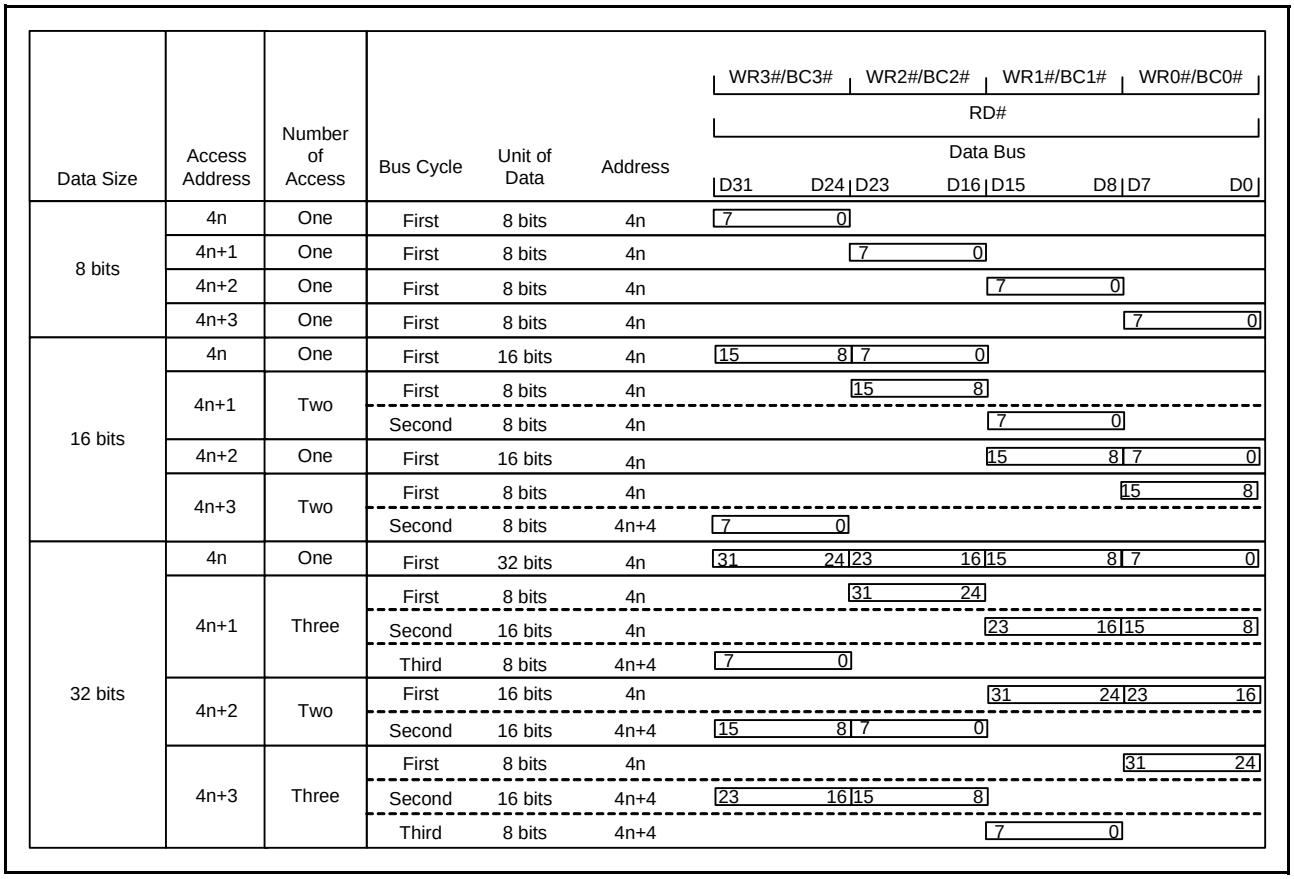
16.4.1 Data Alignment Control for CS Area

(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	WR3#/BC3# | WR2#/BC2# | WR1#/BC1# | WR0#/BC0# / RD# / Data Bus: D31–D24 | D23–D16 | D15–D8 | 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	7–0
	4n+3	One	First	8 bits	4n	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	15–8
	4n+2	One	First	16 bits	4n	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	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	Three	First	8 bits	4n	7–0
			Second	16 bits	4n+4	23–16, 15–8
			Third	8 bits	4n+4	31–24

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 WR2# and WR3# 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# D15–D8	WR0#/BC0# D7–D0
							RD# Data Bus	
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# D15–D8	WR0#/BC0# 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		Data Bus (WR1#/BC1# D15–D8, WR0#/BC0# D7–D0, RD#)
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

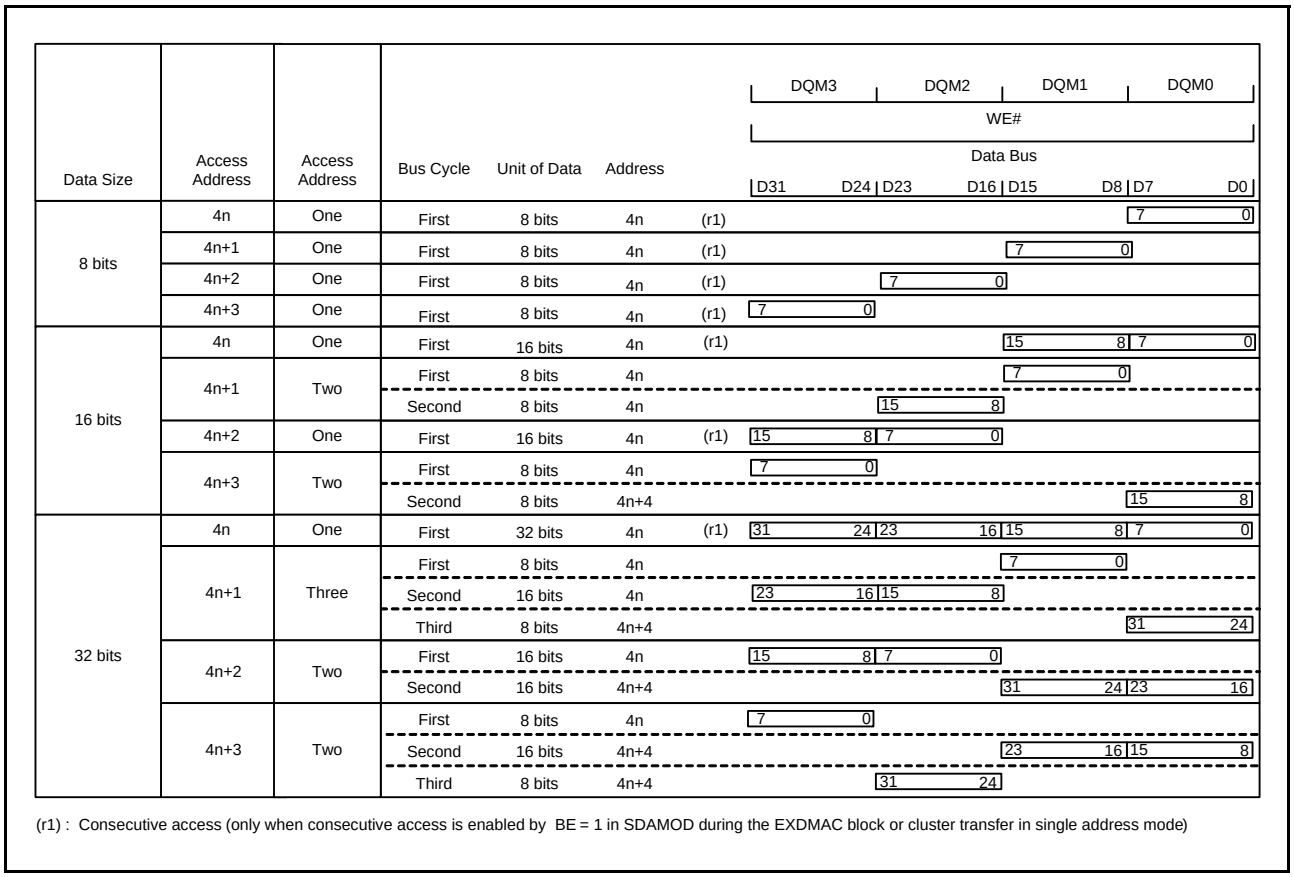
(1) 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A27 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	
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 16.12 Data Alignment (Little Endian) in 32-Bit Bus Space

Data Size	Access Address	Number of Access	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		15 8
			Second	8 bits	4n		7 0
	4n+2	One	First	16 bits	4n	(r1)	15 8 7 0
	4n+3	Two	First	8 bits	4n		15 8
			Second	8 bits	4n+4		7 0
32 bits	4n	One	First	32 bits	4n	(r1)	31 24 23 16 15 8 7 0
	4n+1	Three	First	8 bits	4n		31 24
			Second	16 bits	4n		23 16 15 8
			Third	8 bits	4n+4		7 0
	4n+2	Two	First	16 bits	4n		31 24 23 16
			Second	16 bits	4n+4		15 8 7 0
	4n+3	Two	First	8 bits	4n		31 24
			Second	16 bits	4n+4		23 16 15 8
			Third	8 bits	4n+4		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 16.13 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 A27 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 A27 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

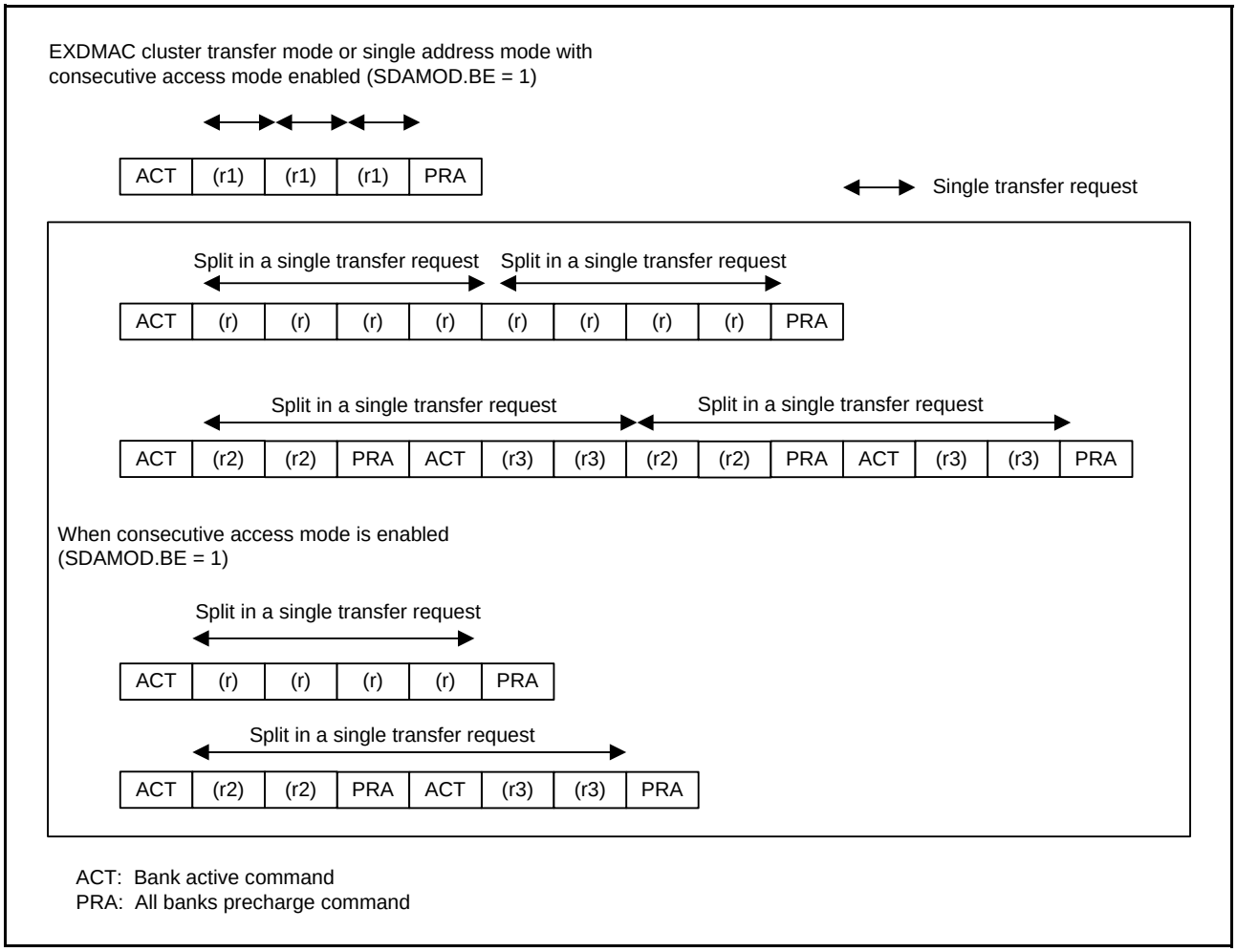


Figure 16.18 Consecutive Access Example

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 (seeFigure 16.27 to Figure 16.31). 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 16.49 and Figure 16.51).

(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 is 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), the RD assert wait select bits (RDON), the WR assert wait select bits (WRON), and the write data output wait select bits (WDON) of 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 5. 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 4. 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) and the write-access CS extension cycle select bits (CSWOFF) in the CSn wait control register 2 (CSnWCR2), respectively.
The number of cycles is 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 3. 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 3. 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.

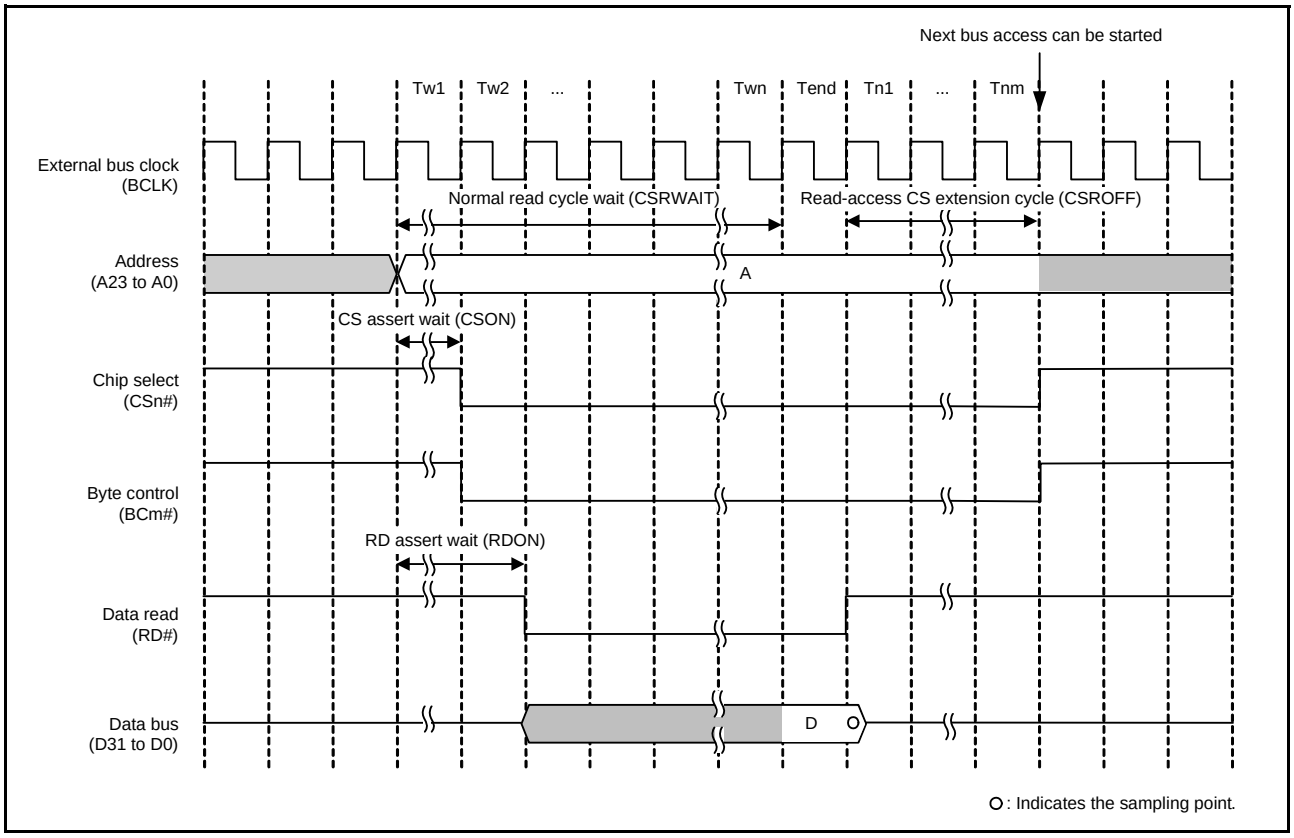


Figure 16.19 Bus Timing (Normal-Read Operation) (n = 0 to 7, m = 0 to 3)

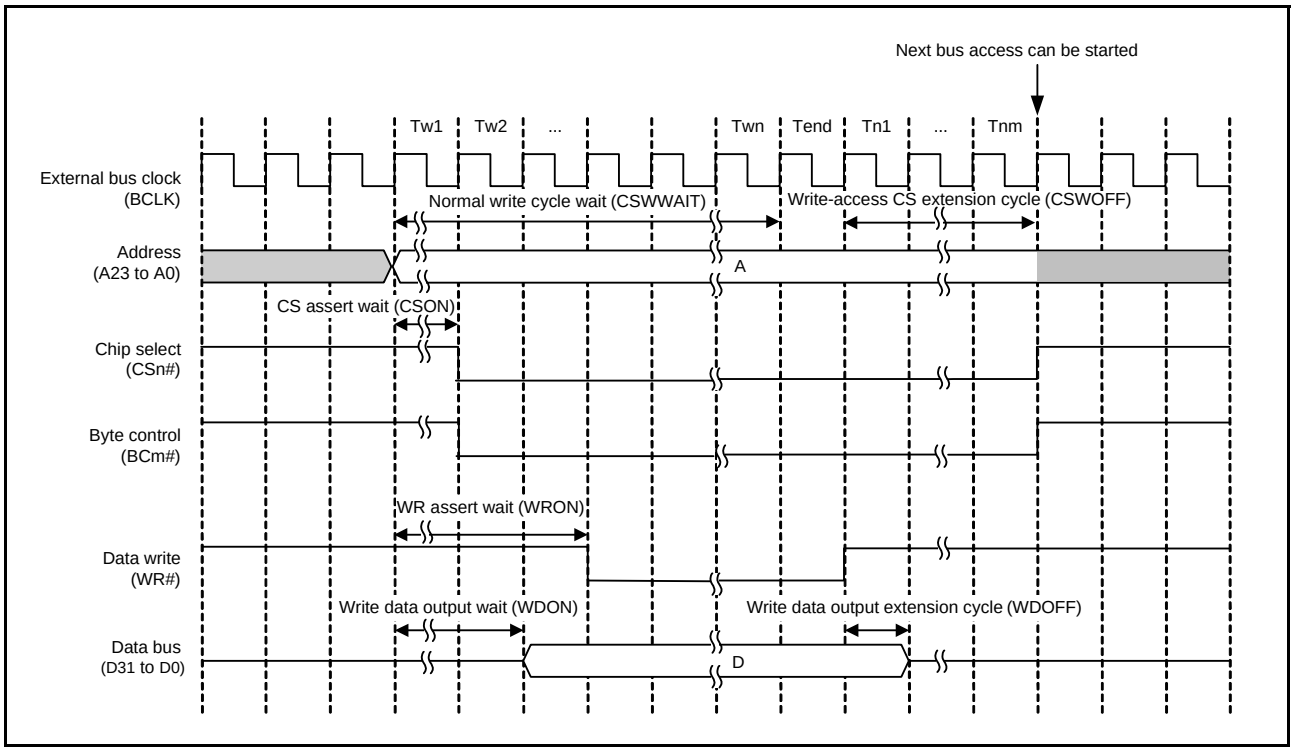


Figure 16.20 Bus Timing (Normal-Write Operation, Single Write Strobe Mode) (n = 0 to 7, m = 0 to 3)

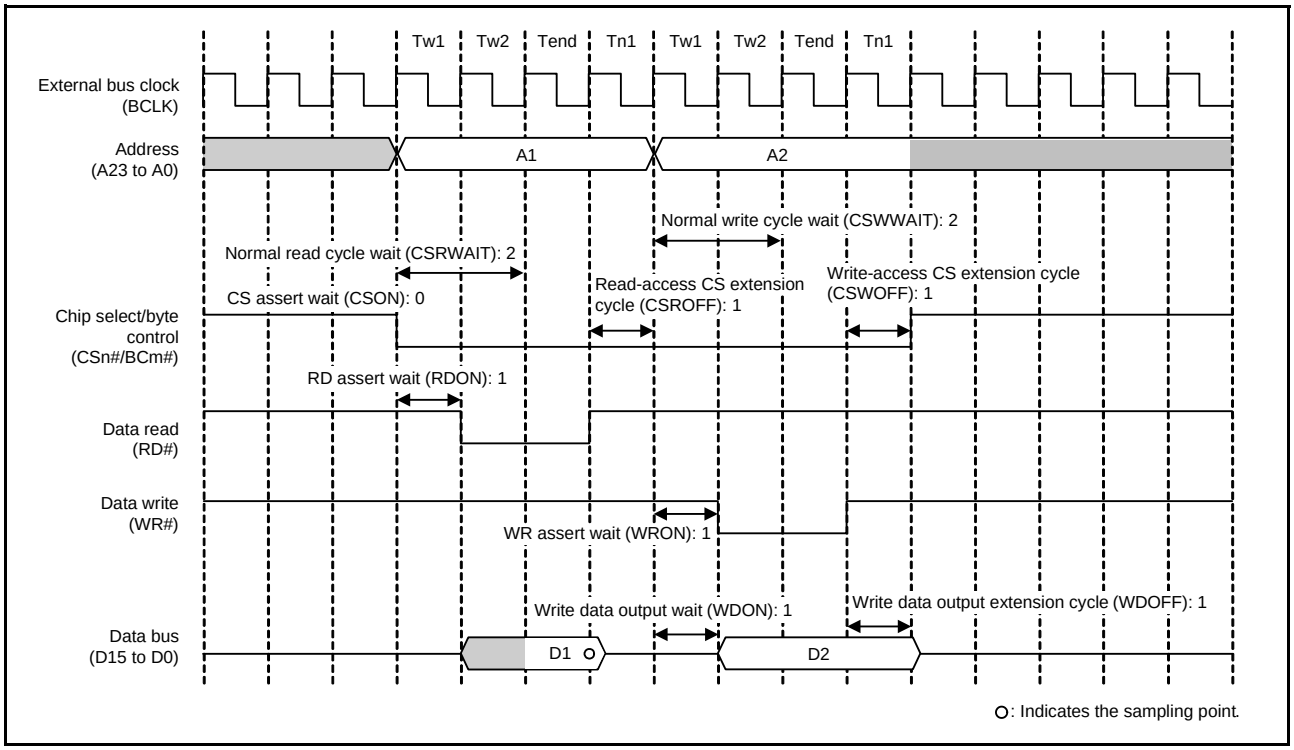


Figure 16.21 Example of Normal Access Operation (Read/Write) (n = 0 to 7, m = 0 to 3)

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 16.47).
The values of the wait control registers are example settings. In practice, set the register bits according to the specifications of connected devices.

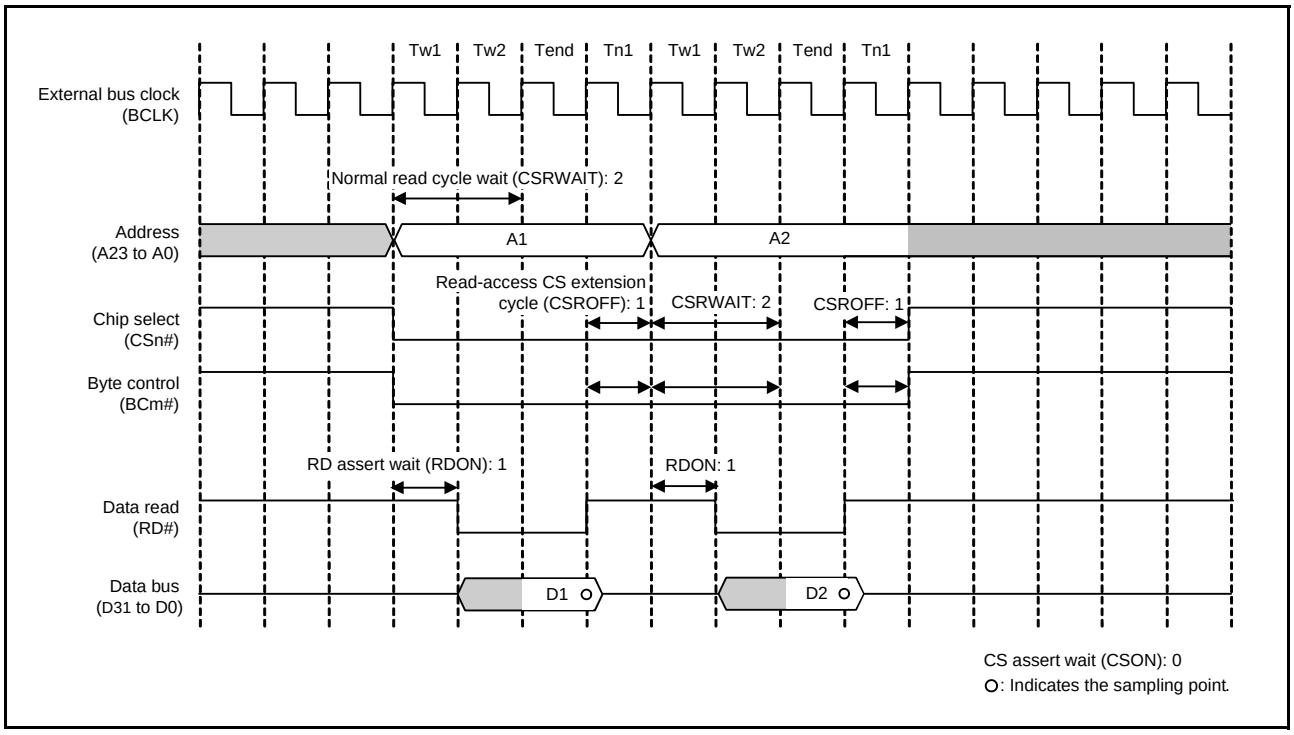


Figure 16.22 Example of Normal-Read Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

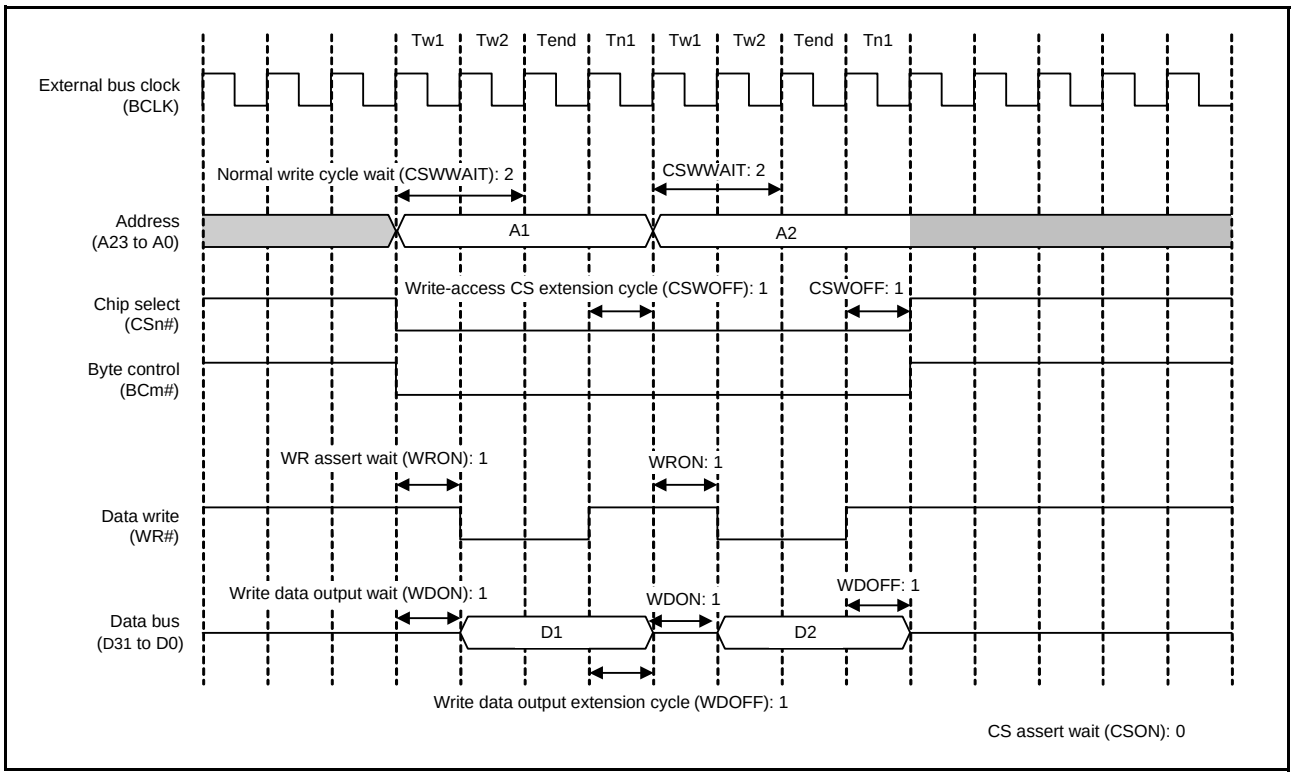


Figure 16.23 Example of Normal-Write Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7, m = 0 to 3)

Figure 16.24 and Figure 16.25 show examples of normal read and write accesses to a 32-bit bus space in 16 bits.

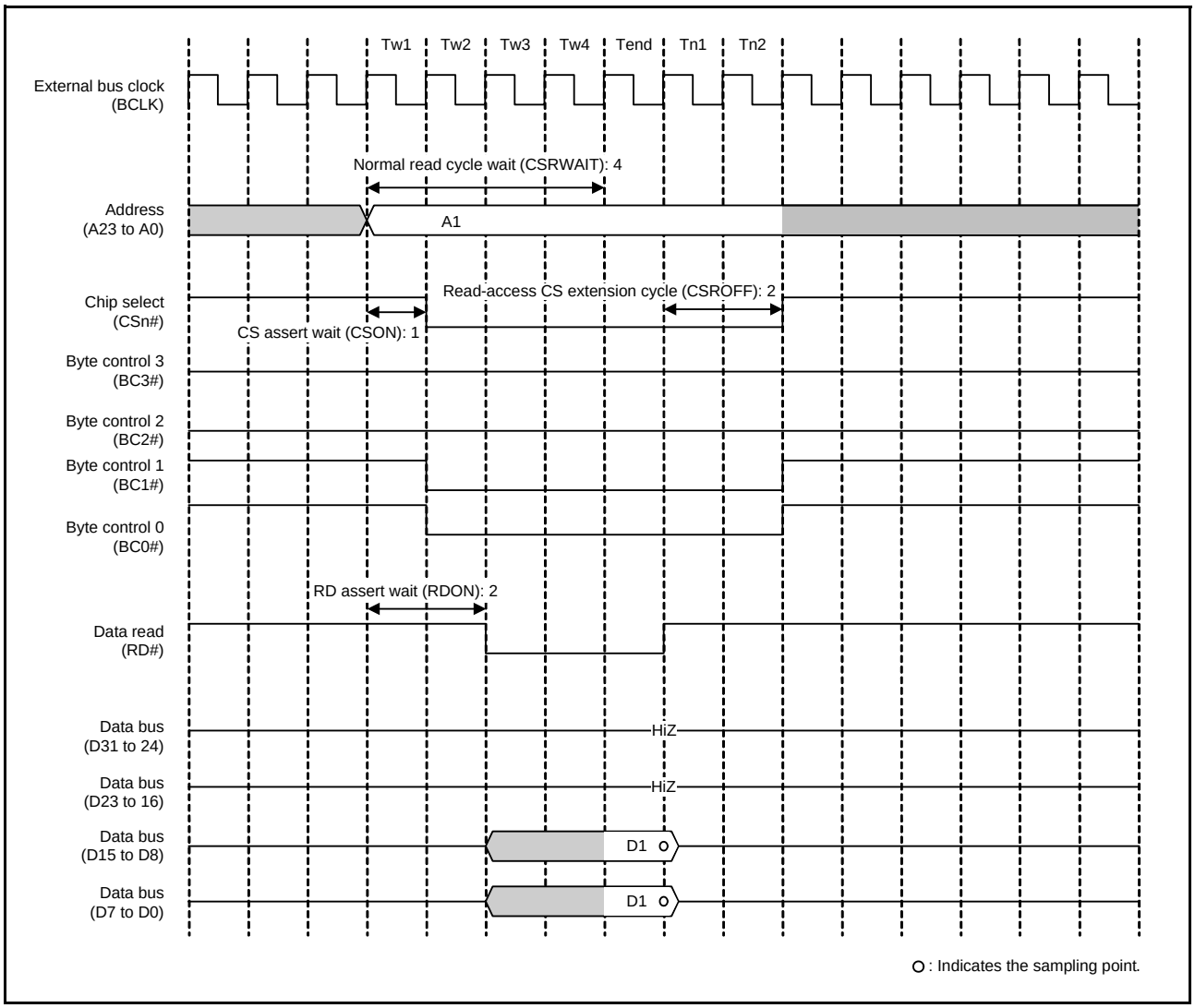


Figure 16.24 Example of Normal-Read Operation (when 32-Bit Bus Space is Accessed in 16 Bits) (n = 0 to 7)

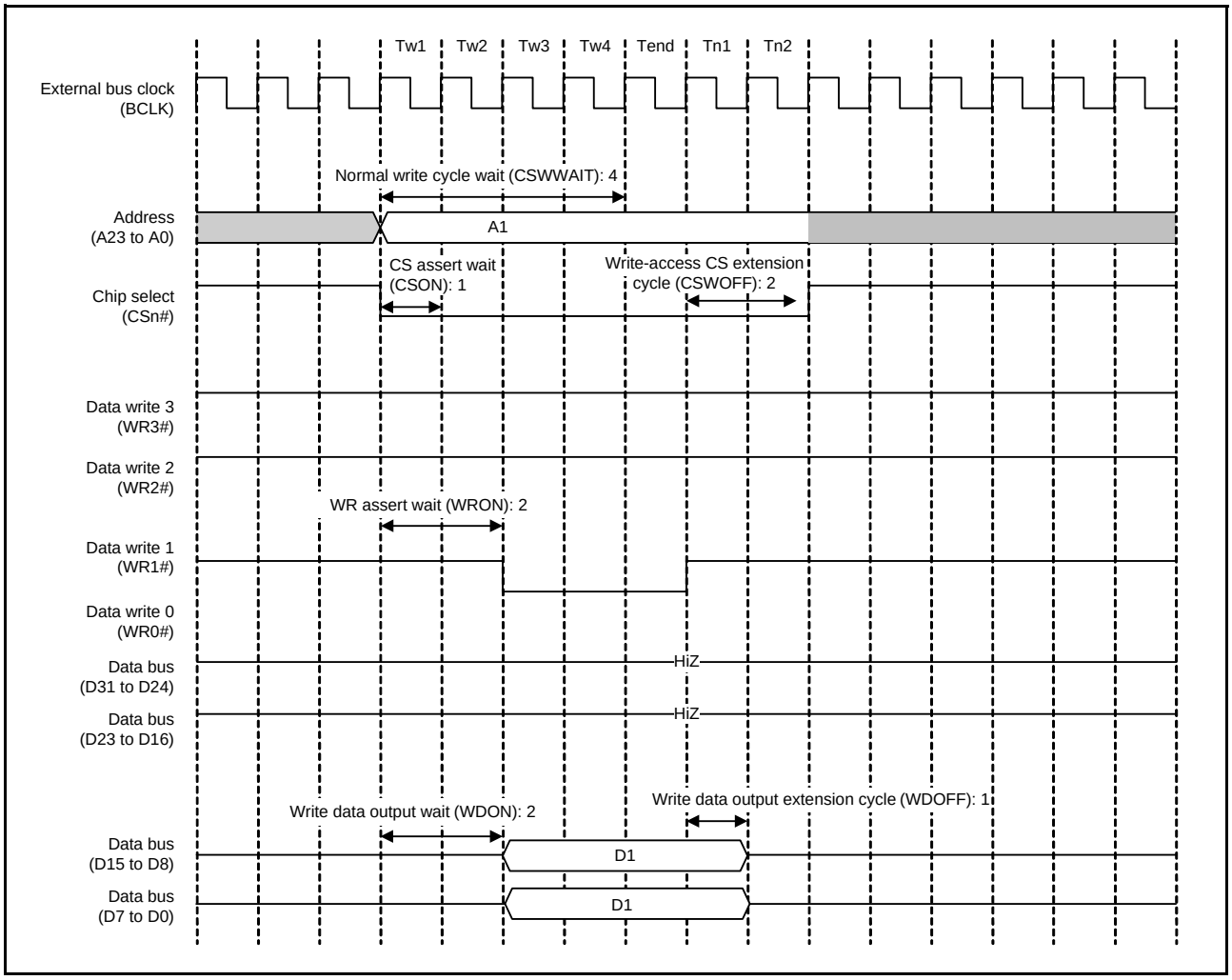


Figure 16.25 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 16 Bits, in Byte Strobe Mode) (n = 0 to 7)

Figure 16.26 shows an example of normal write operation in single write strobe mode.

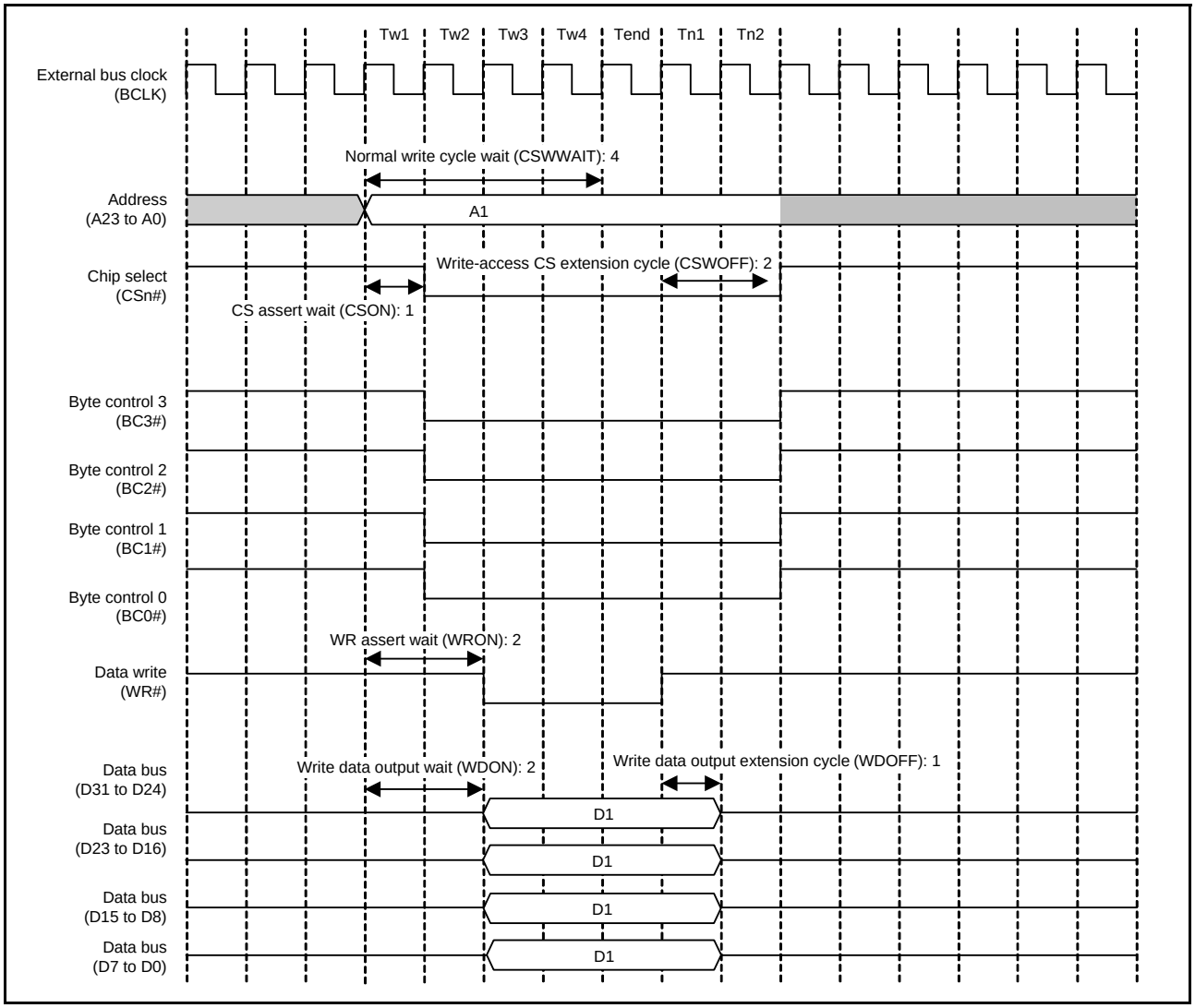


Figure 16.26 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode) (n = 0 to 7)

Figure 16.27 to Figure 16.31 show examples of normal accesses made with the 1/2 BCLK selected with the BCLK pin output select bit.

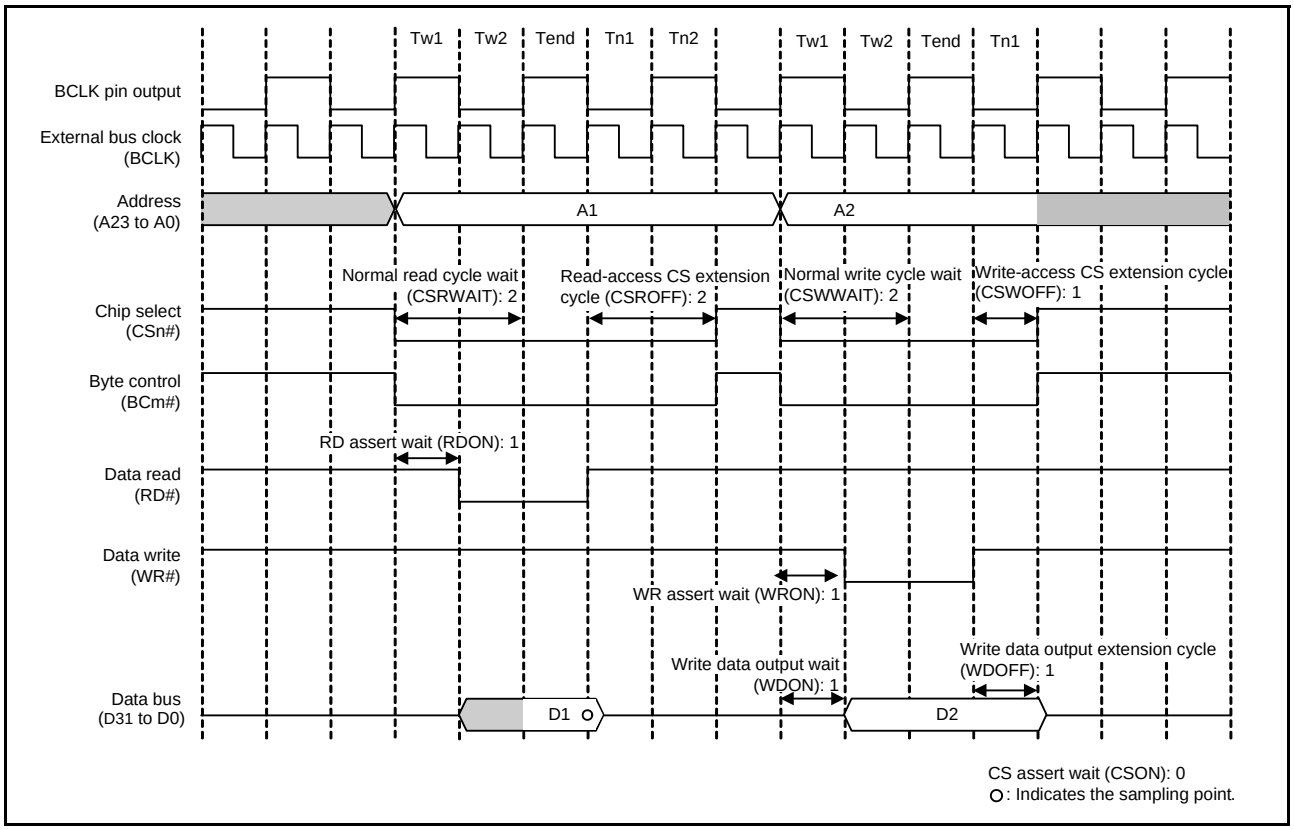


Figure 16.27 Example of Normal Access (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

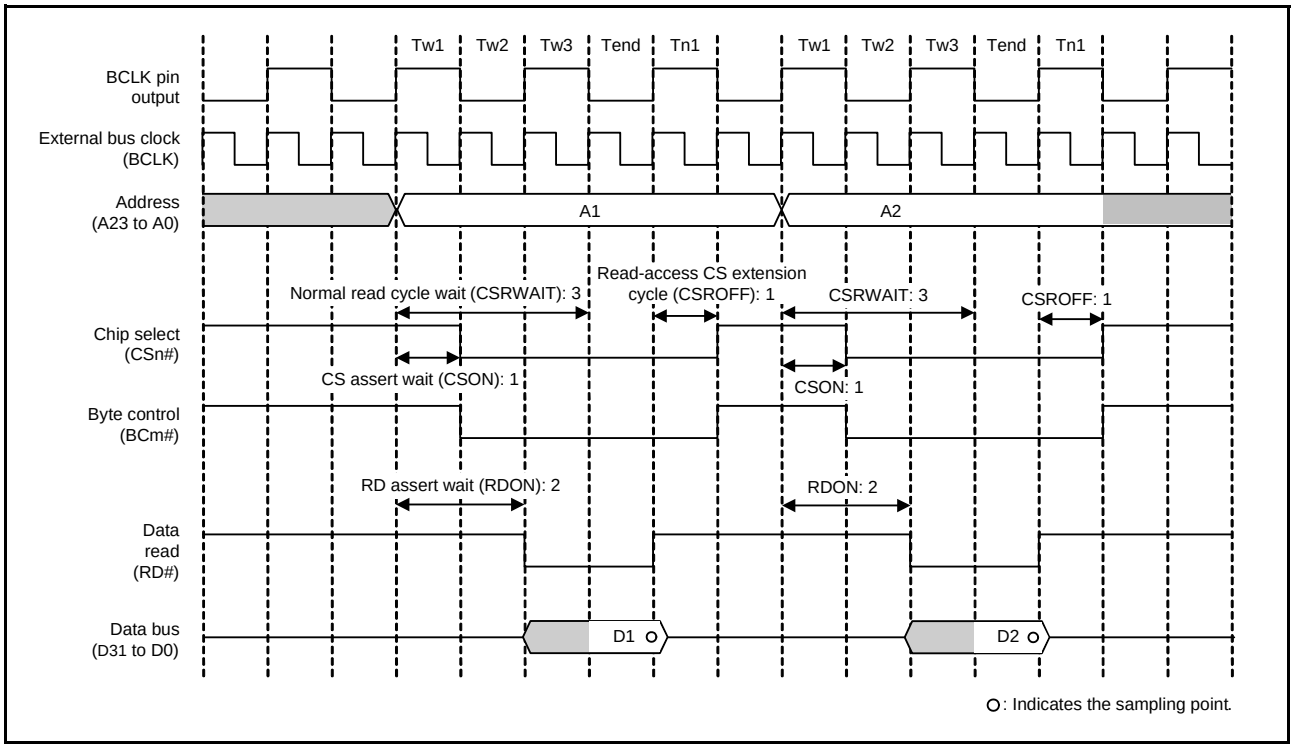


Figure 16.28 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

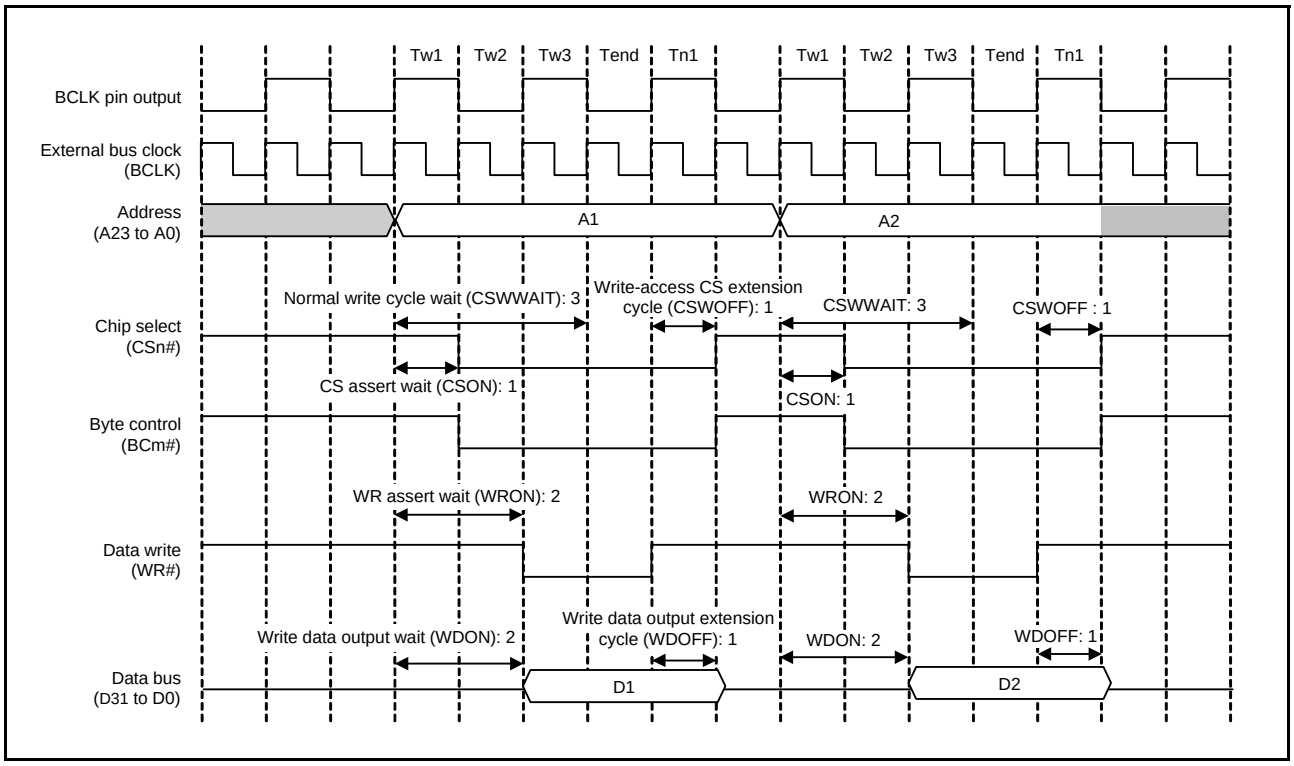


Figure 16.29 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

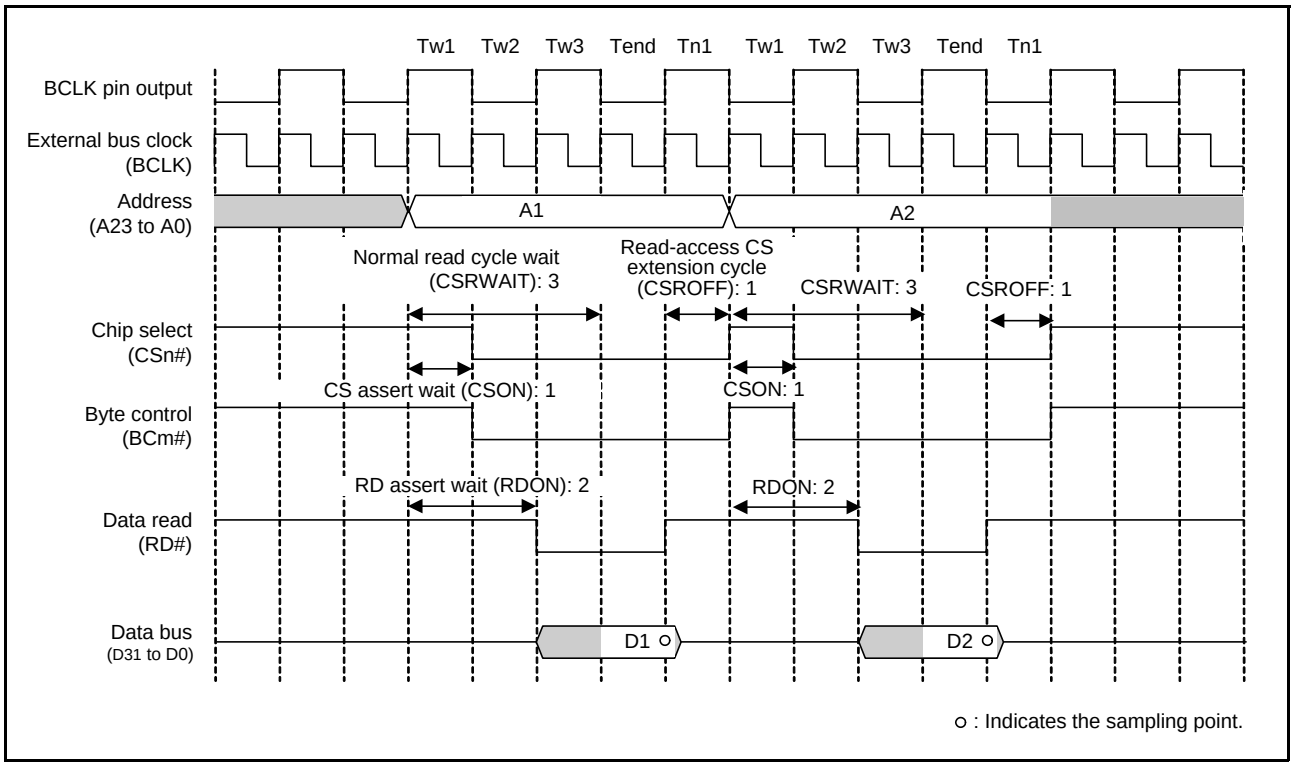


Figure 16.30 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

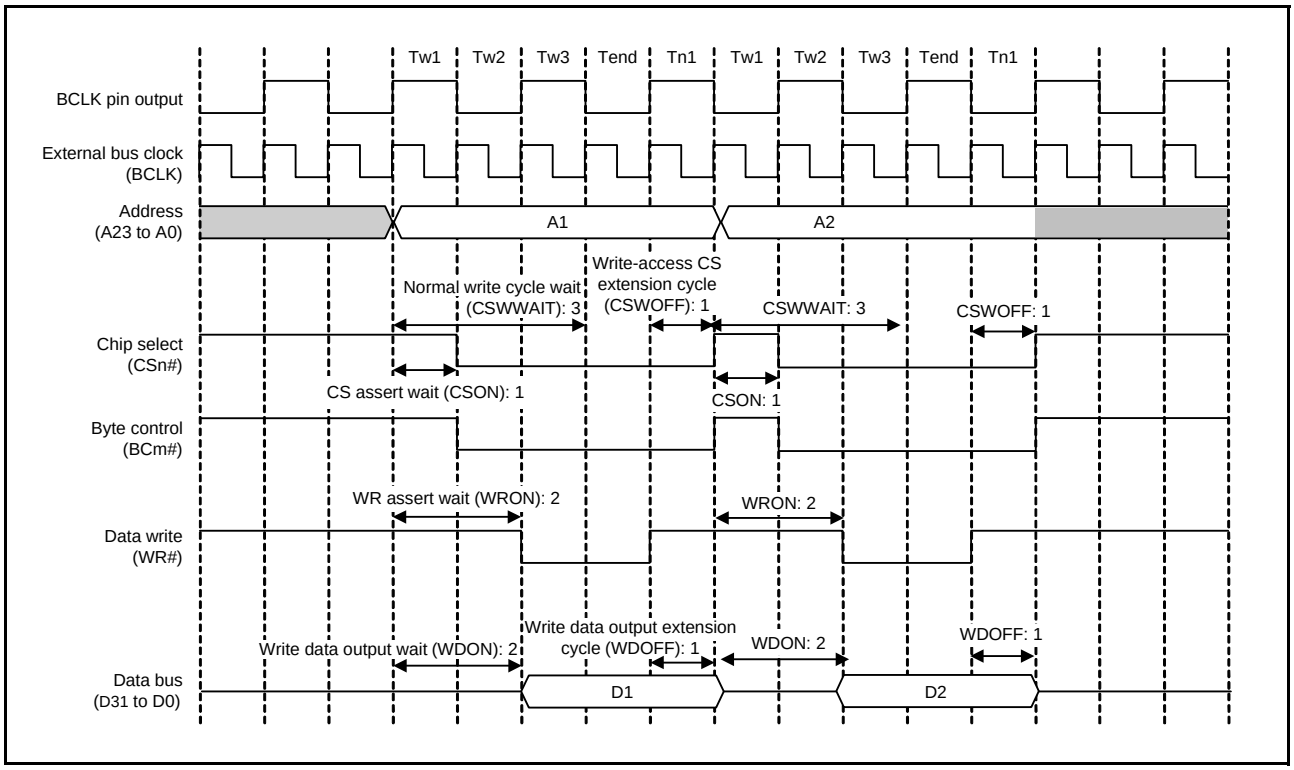


Figure 16.31 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

(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 16.32 and Figure 16.33 show examples of page access operations.

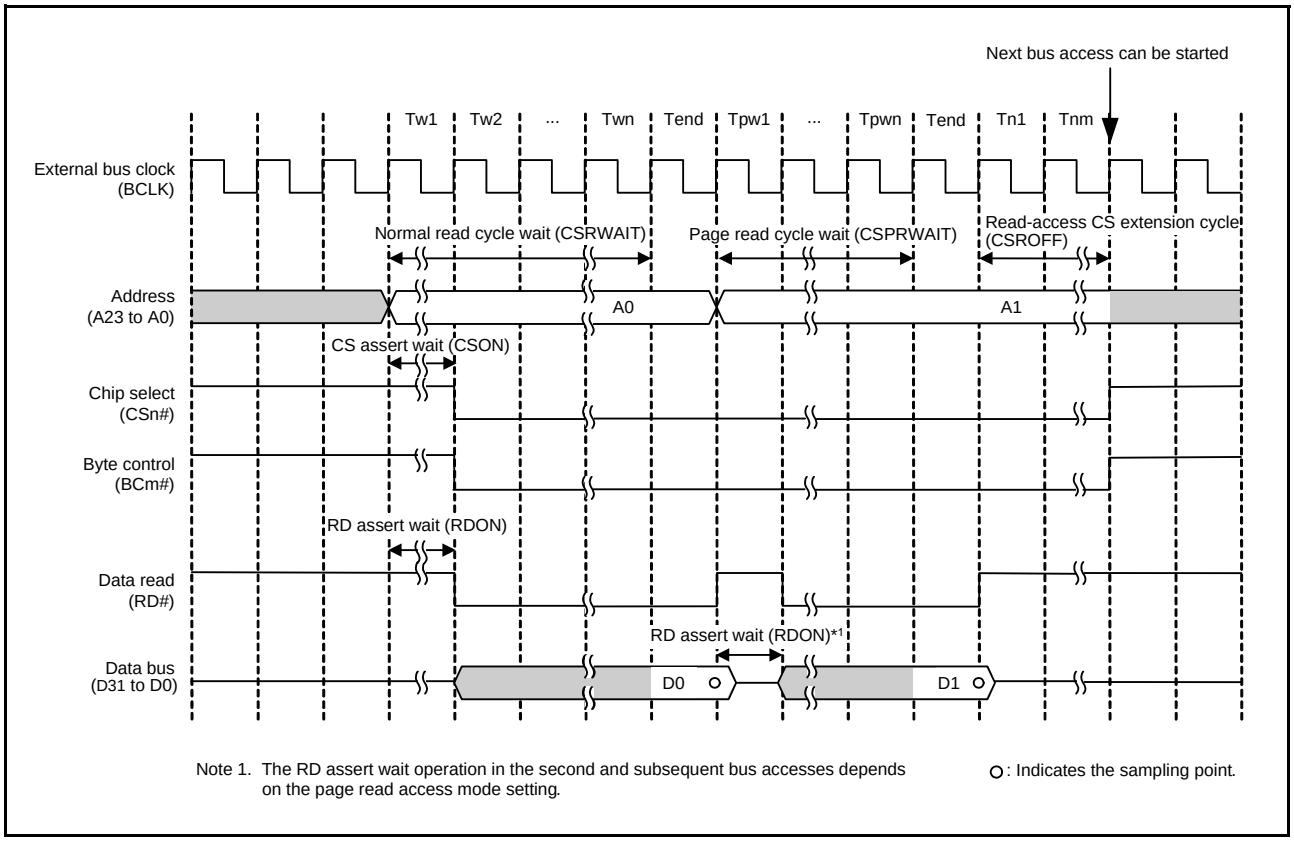


Figure 16.32 Page-Read Access Timing (n = 0 to 7, m = 0 to 3)

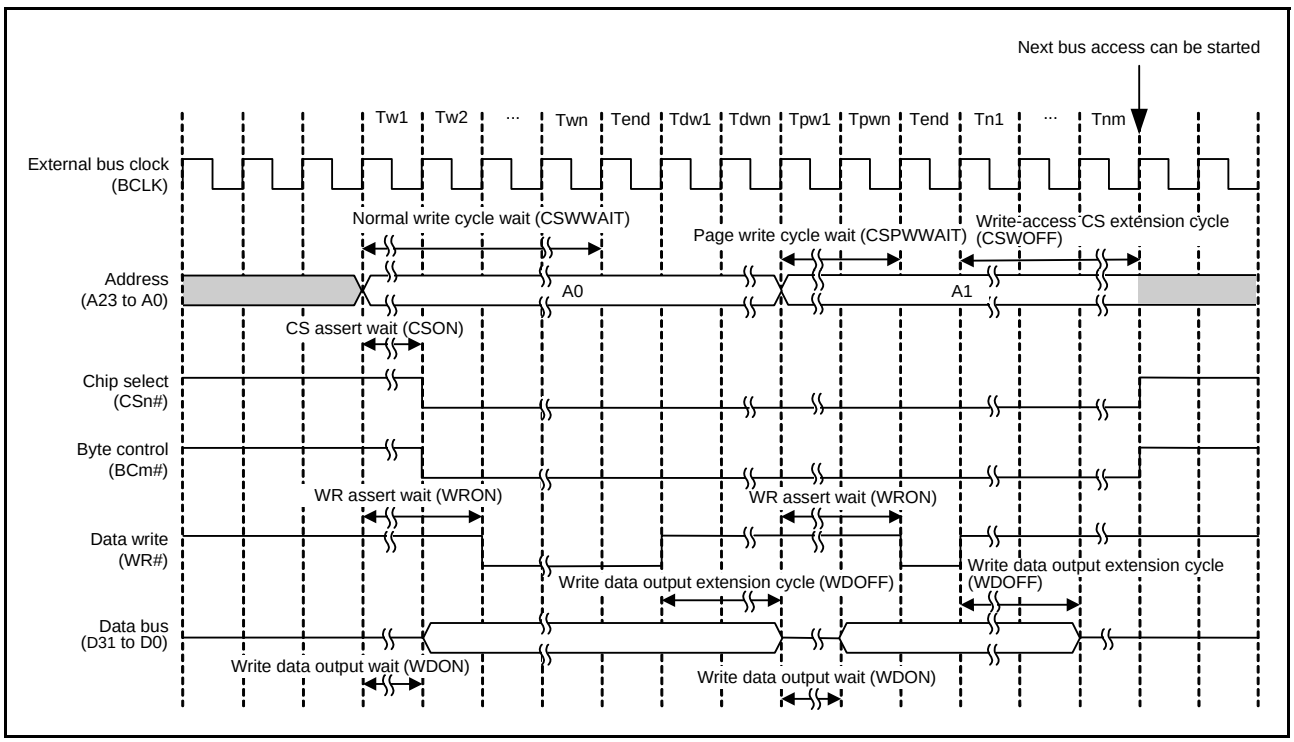


Figure 16.33 Page-Write Access Timing (n = 0 to 7, m = 0 to 3)

Figure 16.34 and Figure 16.35 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.

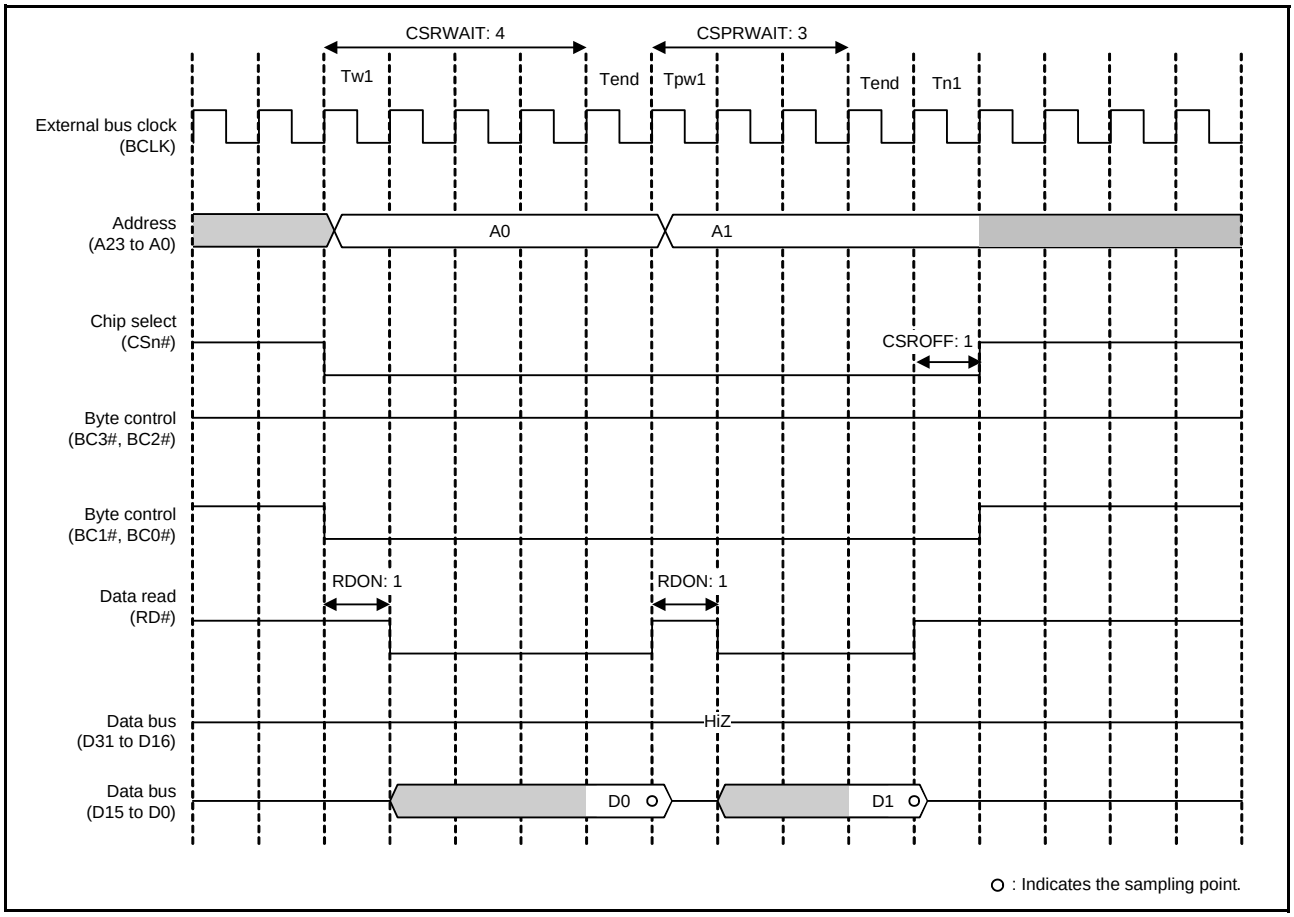


Figure 16.34 Example of Page-Read Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits) (n = 0 to 7)

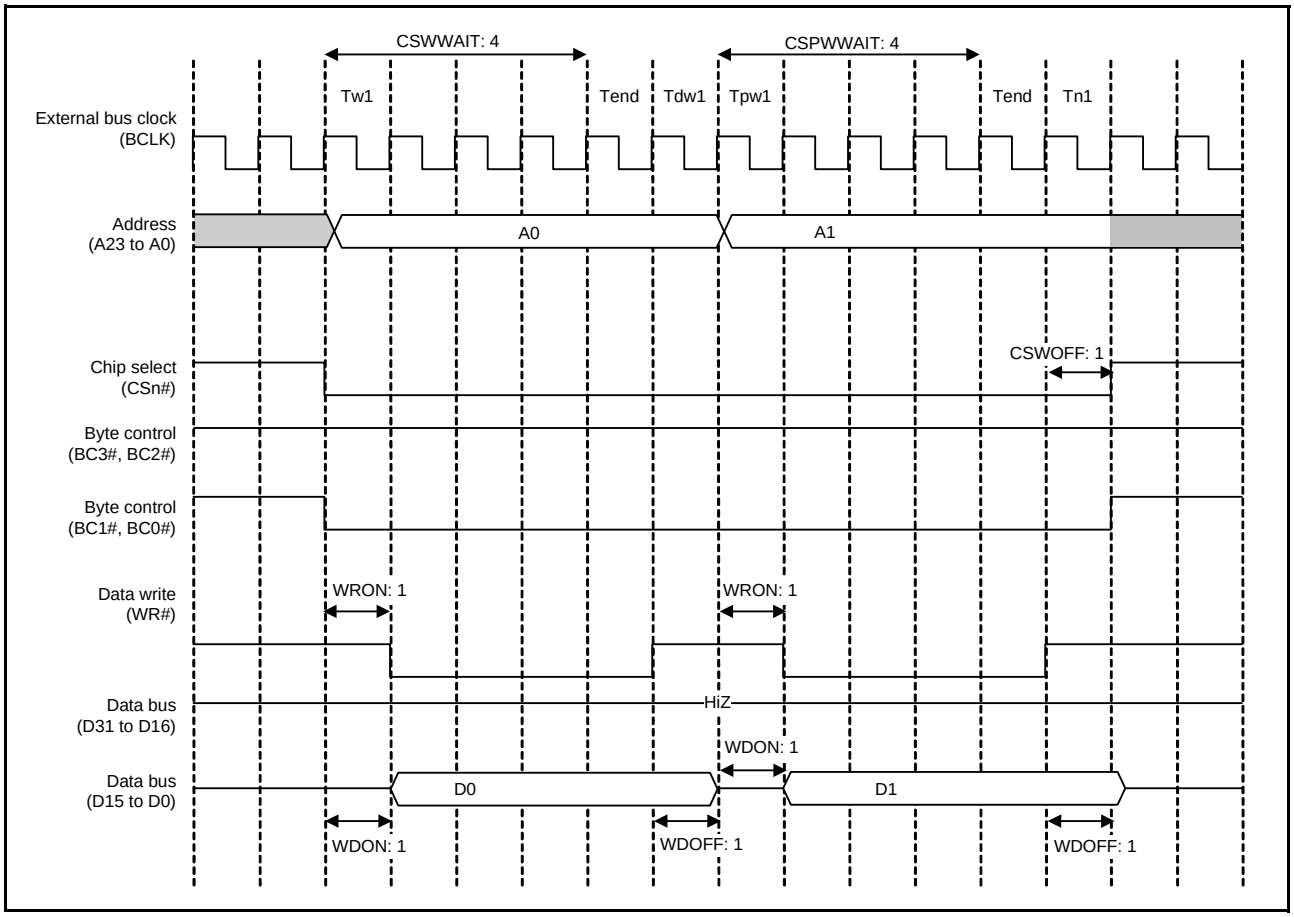


Figure 16.35 Example of Page-Write Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode) (n = 0 to 7)

Figure 16.36 and Figure 16.37 show examples of page access operations performed with the 1/2 BCLK selected with the BCLK pin output select bit.

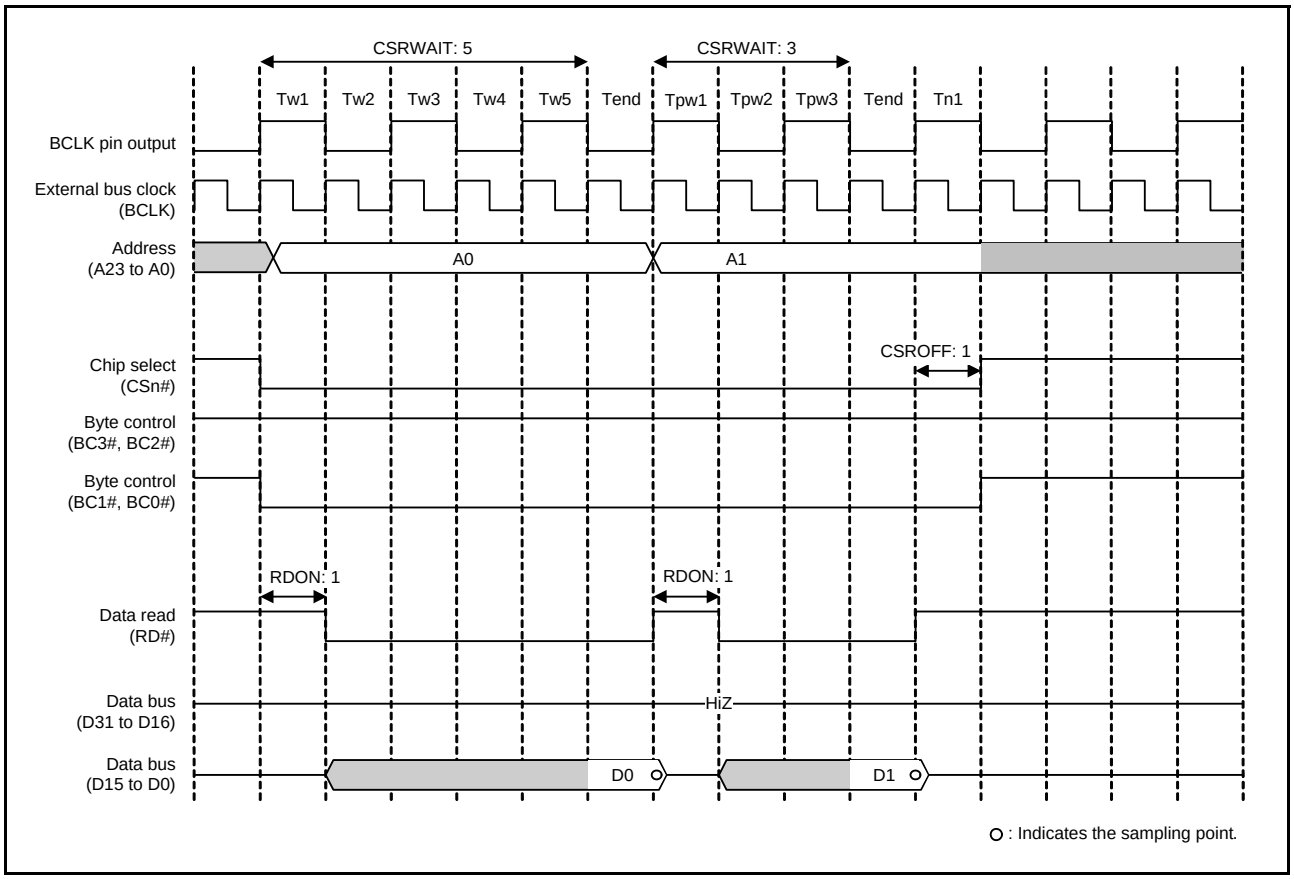


Figure 16.36 Example of Page Read Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7)

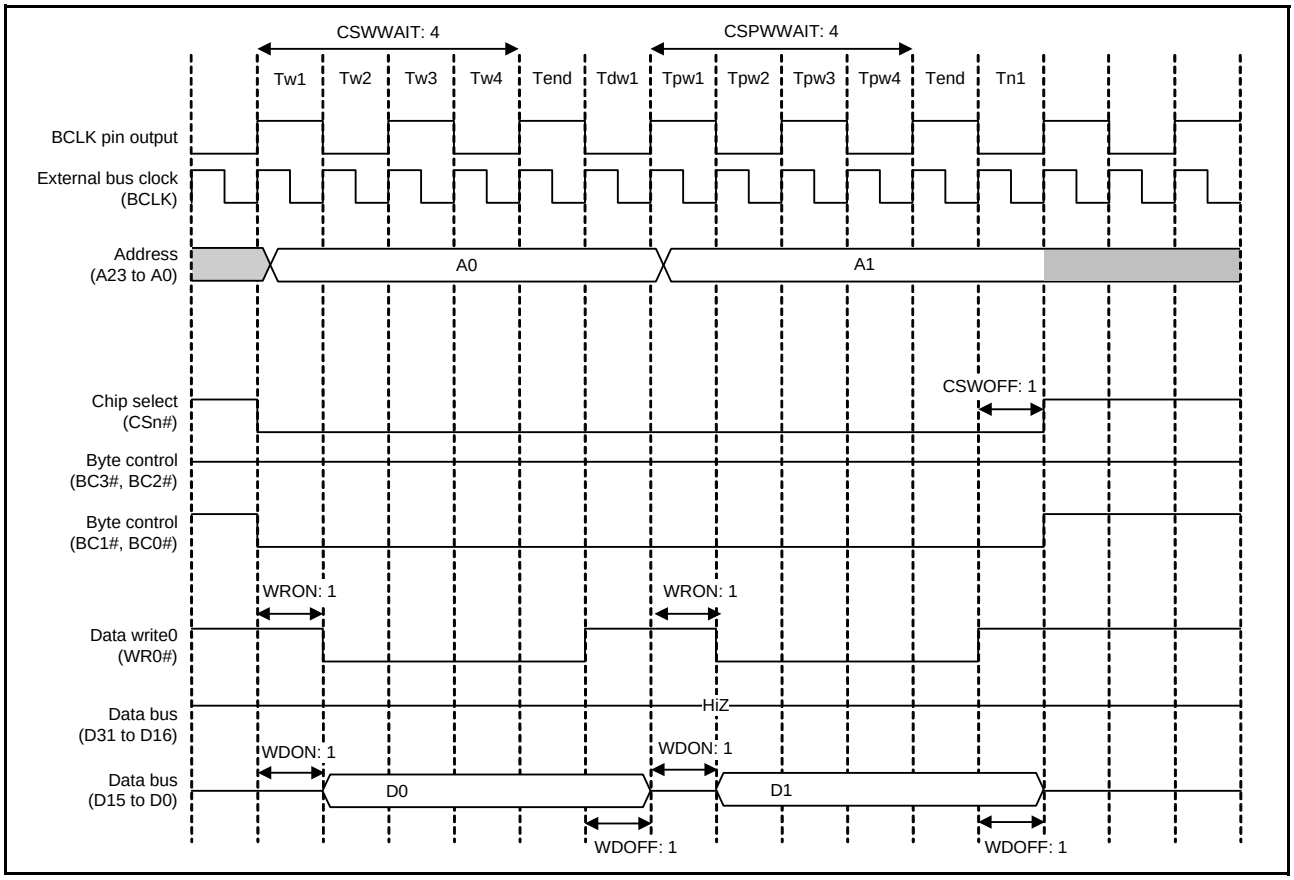


Figure 16.37 Example of Page Write Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7)

16.5.2 Address/Data Multiplexed Bus

When the address/data multiplexed I/O interface select bit (MPXEN) in CS1CR is set to 1, addresses and data can be multiplexing ly 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 CS1CR, 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 CS1WCR2 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 is selectable within the range from zero to 3. 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. CS1WCR1 and CS1WCR2 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 CS1MOD 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 16.38 to Figure 16.40 show examples of operations with the address/data multiplexed I/O interface.

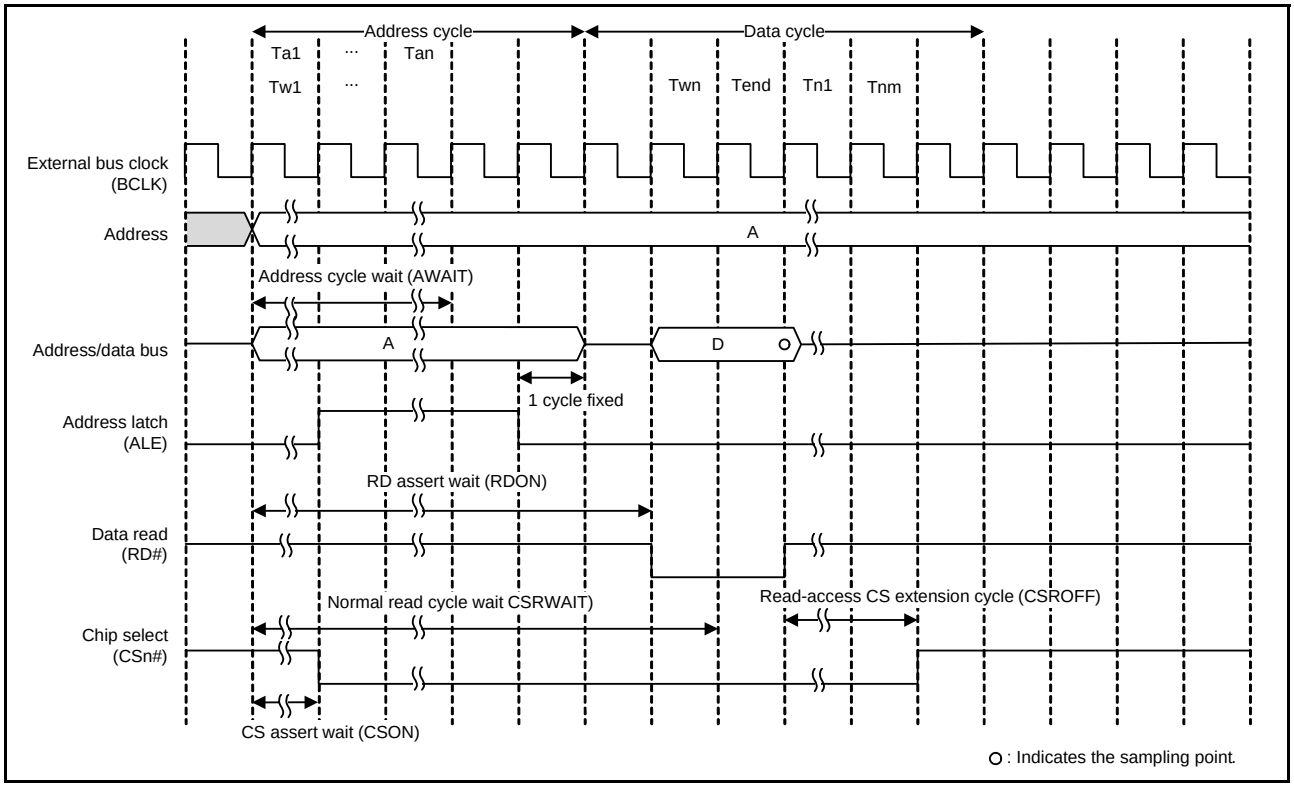


Figure 16.38 Example of Read Access Operation (with Address/Data Multiplexed I/O Interface)

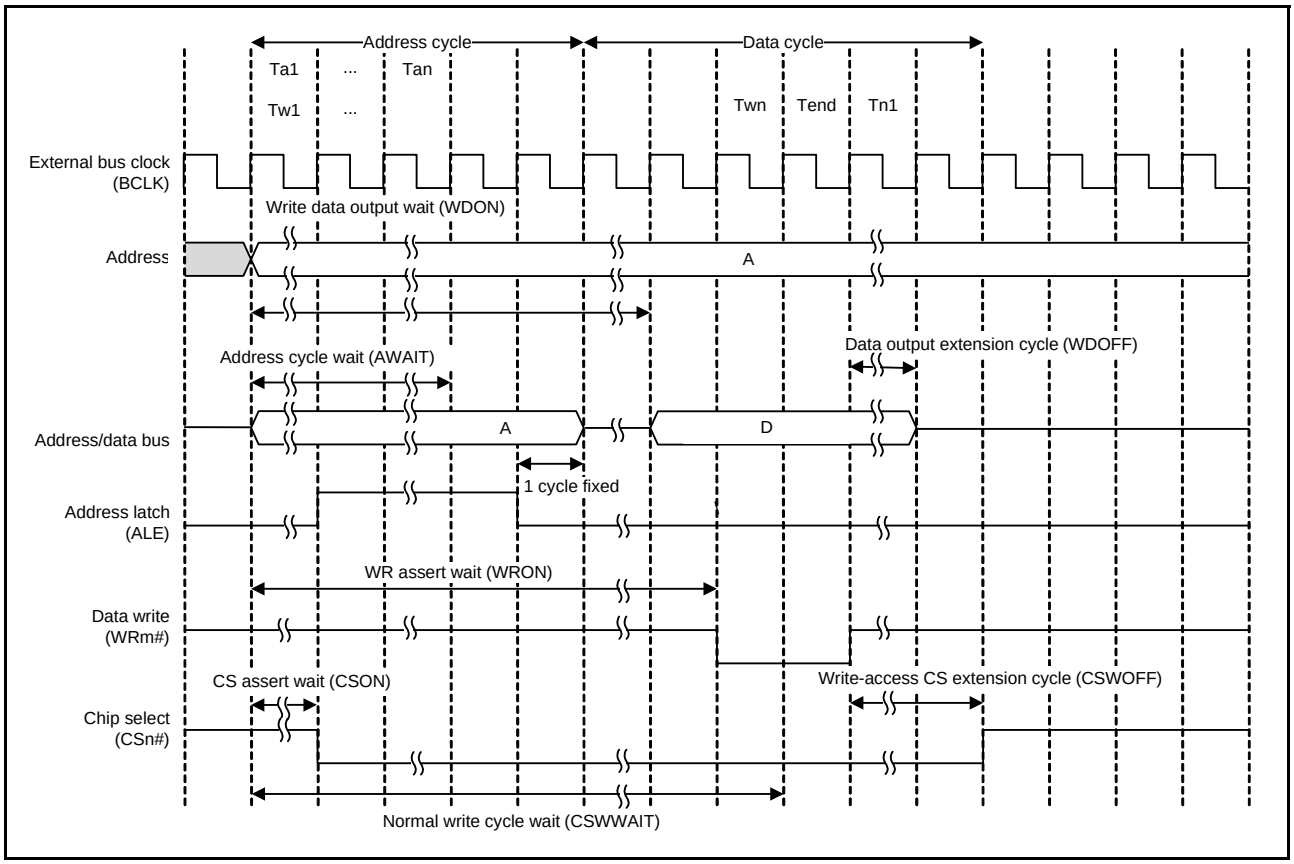


Figure 16.39 Example of Write Access Operation (with Address/Data Multiplexed I/O Interface; m = 0, 1)

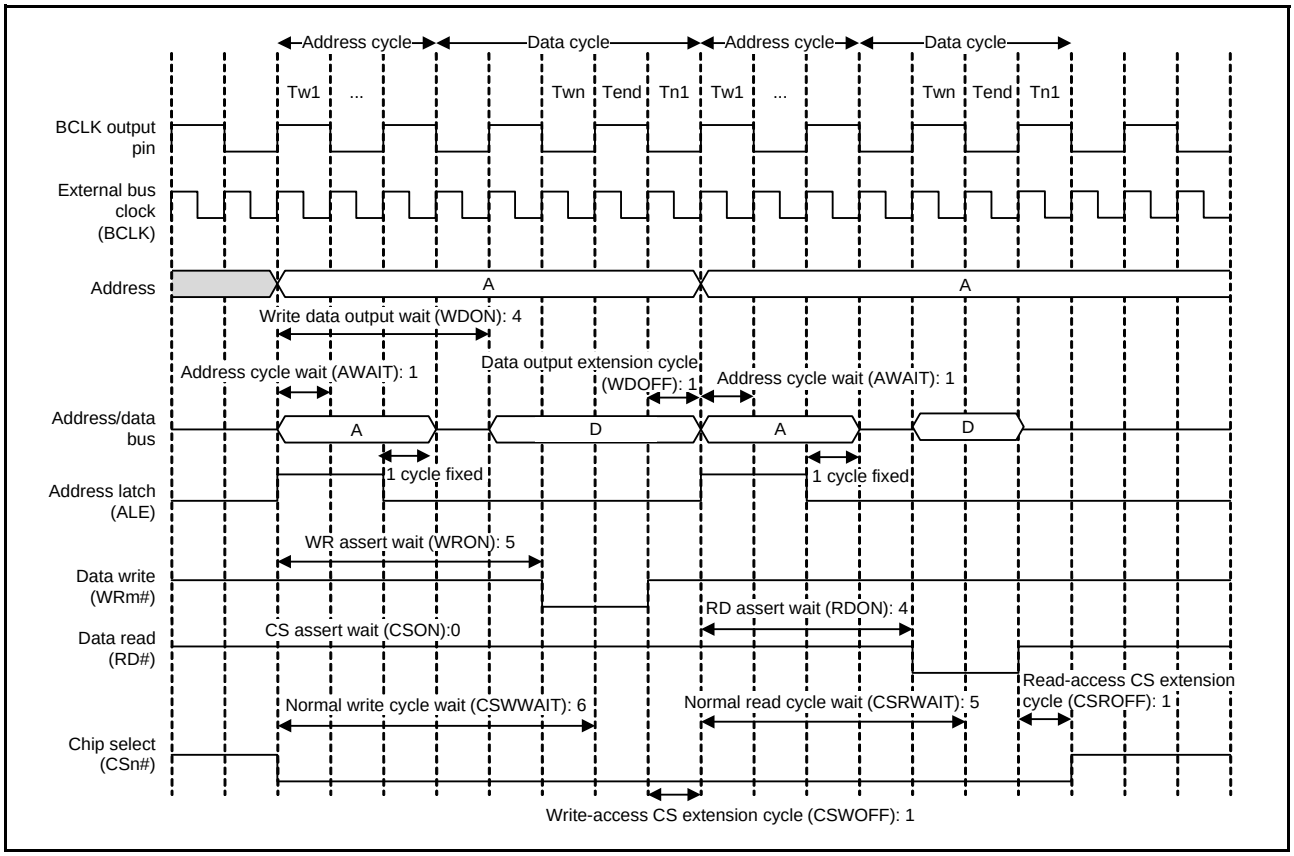


Figure 16.40 Example of Bus Timing (with Address/Data Multiplexed I/O Interface; m = 0, 1)

When two or more rounds of external bus access are required in response to a single transfer request, an address cycle and a data cycle are repeated in the second and subsequent external bus accesses in the same way as the first access operation. For details, see Figure 16.41 and Figure 16.42.

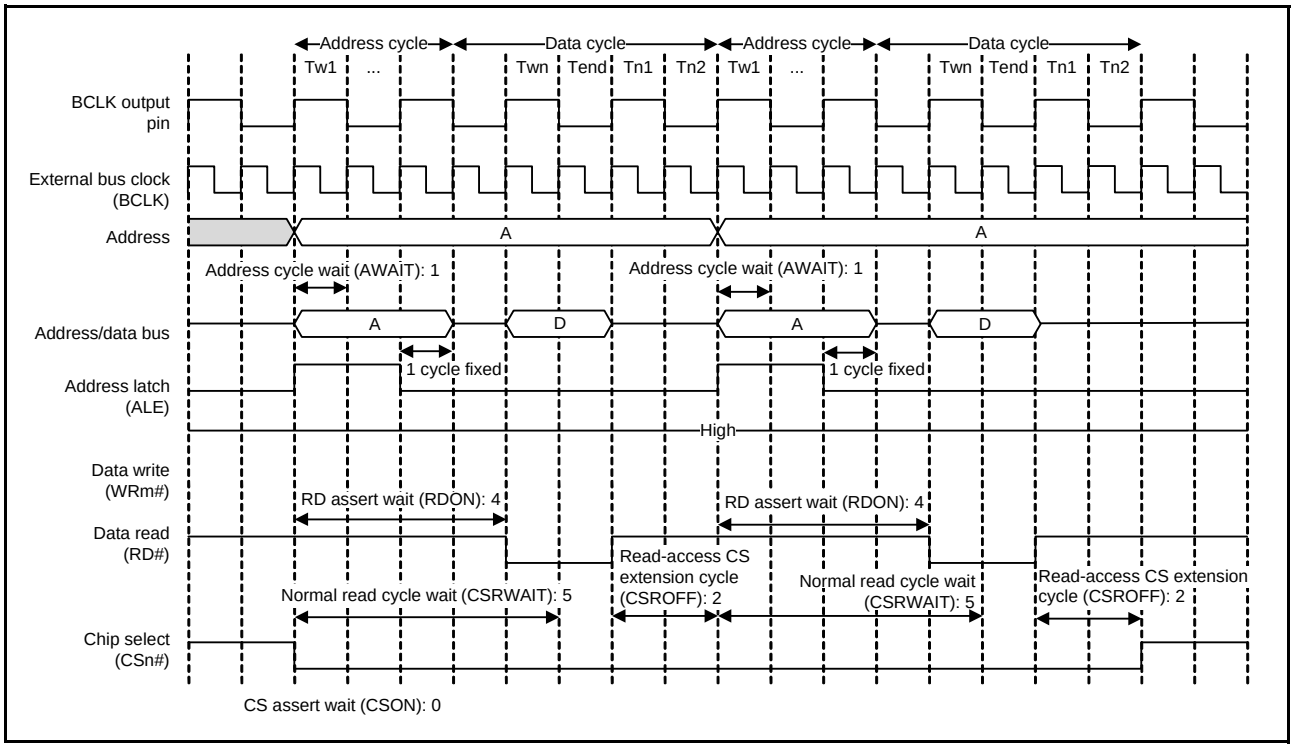


Figure 16.41 Example of Operation When a Read Access is Split (with Address/Data Multiplexed I/O Interface; m = 0, 1)

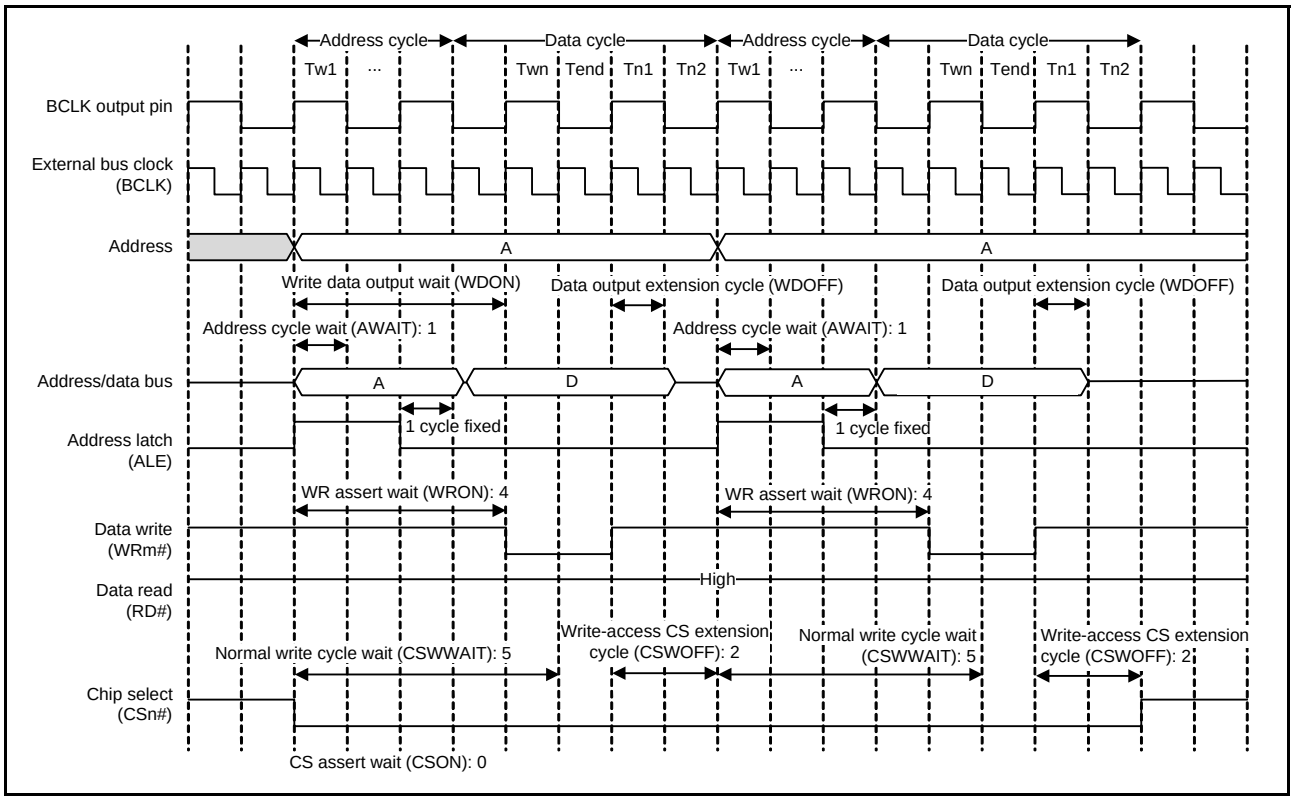


Figure 16.42 Example of Operation When a Write Access is Split (with Address/Data Multiplexed I/O Interface; m = 0, 1)

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 16.43 and Figure 16.44 show examples of external wait insertion timing with the separate bus interface.

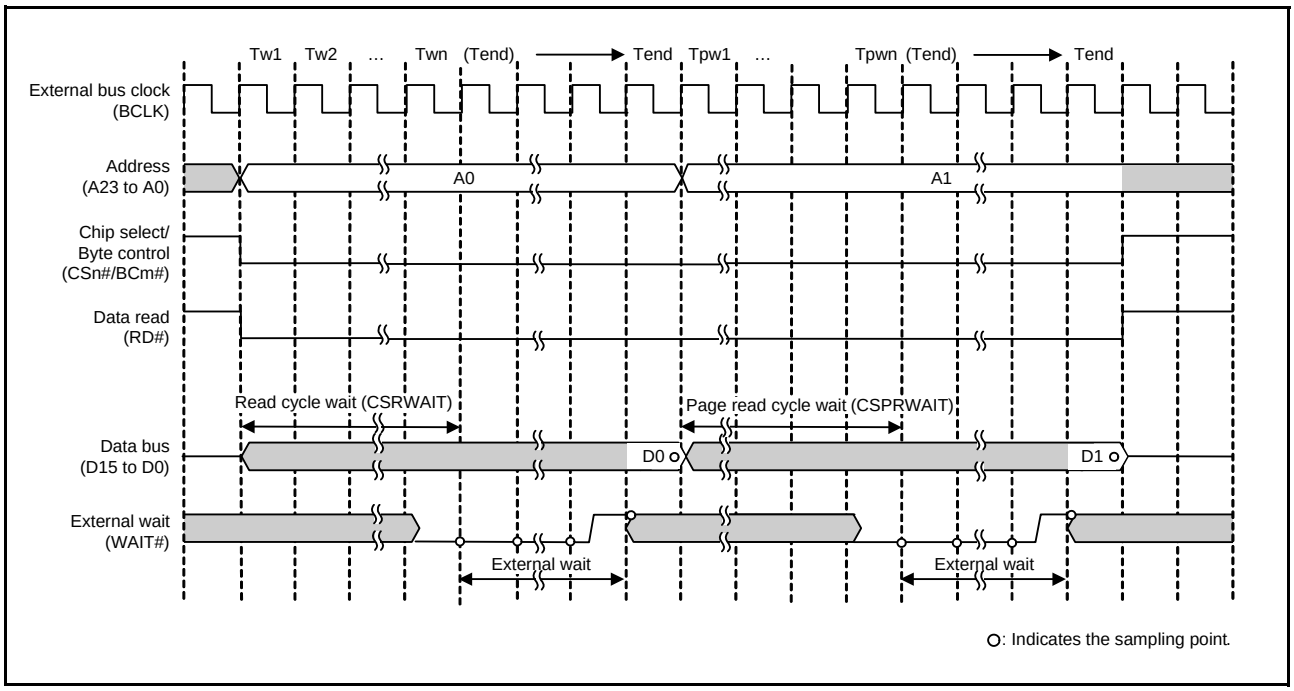


Figure 16.43 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space) (n = 0 to 7, m = 0, 1)

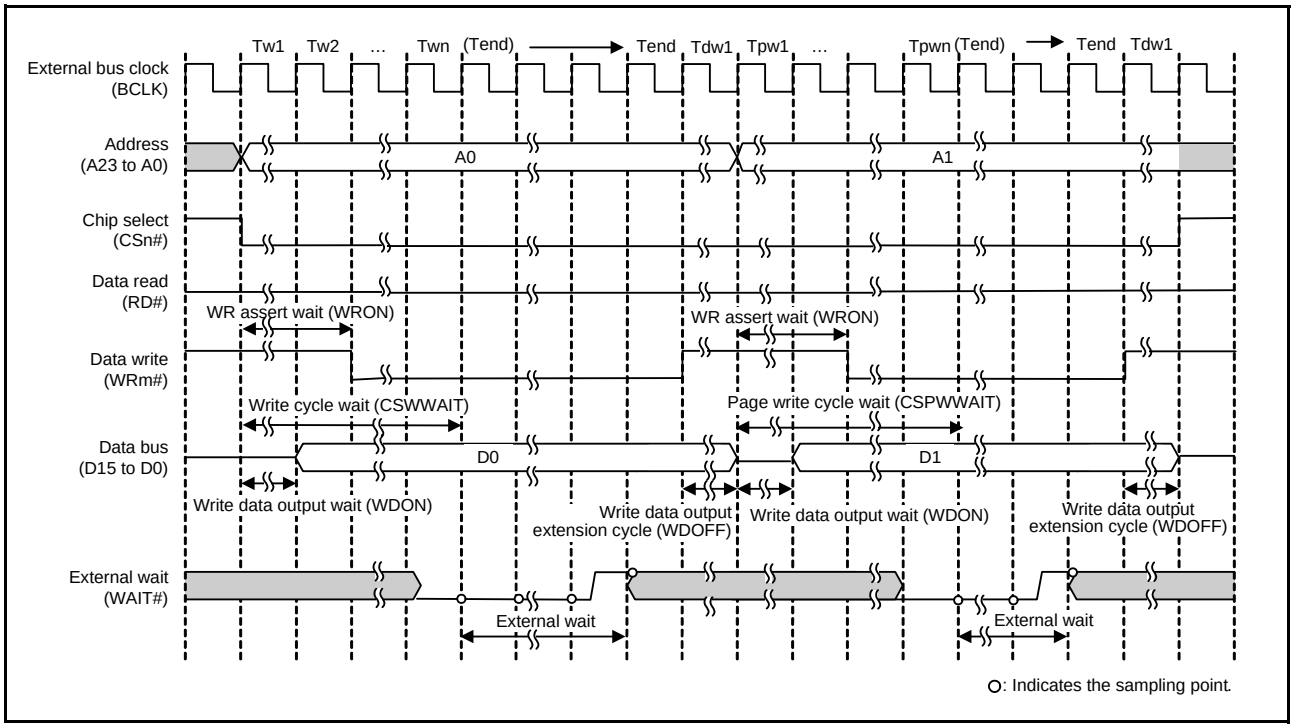


Figure 16.44 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space, in Byte Strobe Mode) (n = 0 to 7, m = 0, 1)

(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 16.45 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

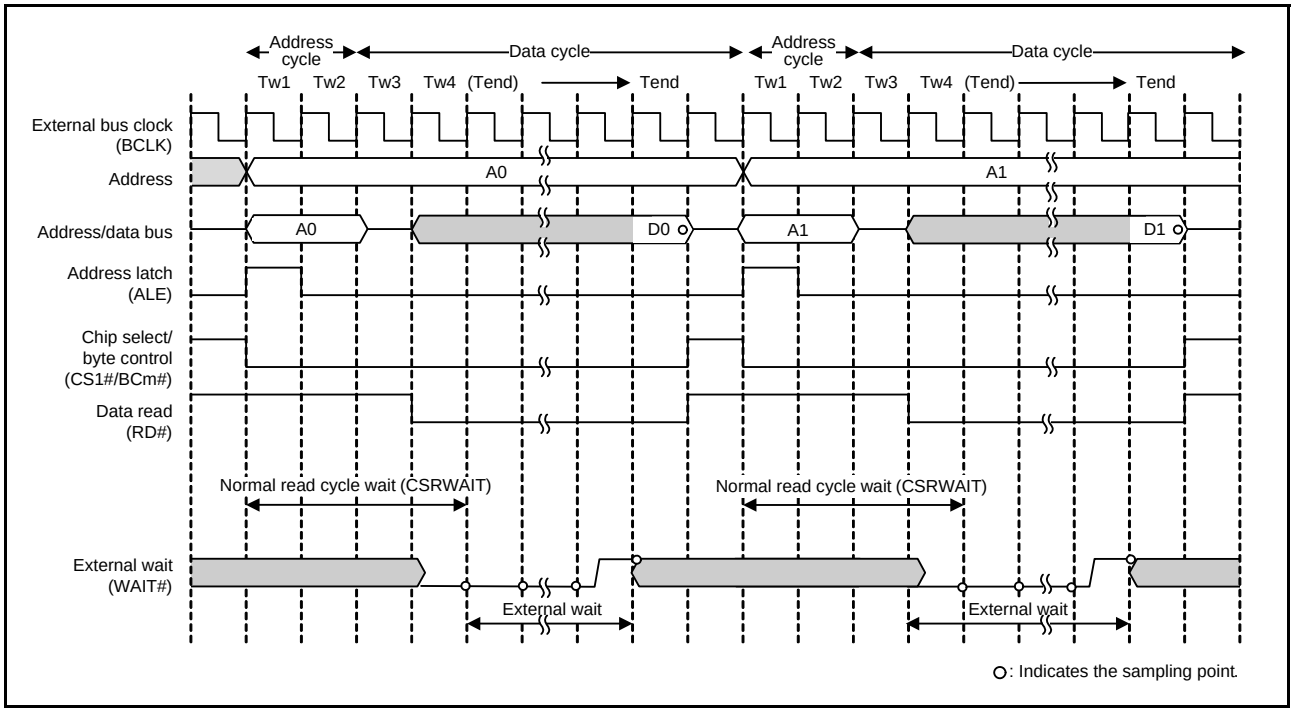


Figure 16.45 Example of External Wait Insertion Timing (with Address/Data Multiplexed I/O Interface; m = 0, 1)

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 16.48). 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. Similarly, during normal accesses with page access enabled, with the separate bus interface, 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 16.46 to Figure 16.48 show examples of recovery cycle insertion with the separate bus interface.

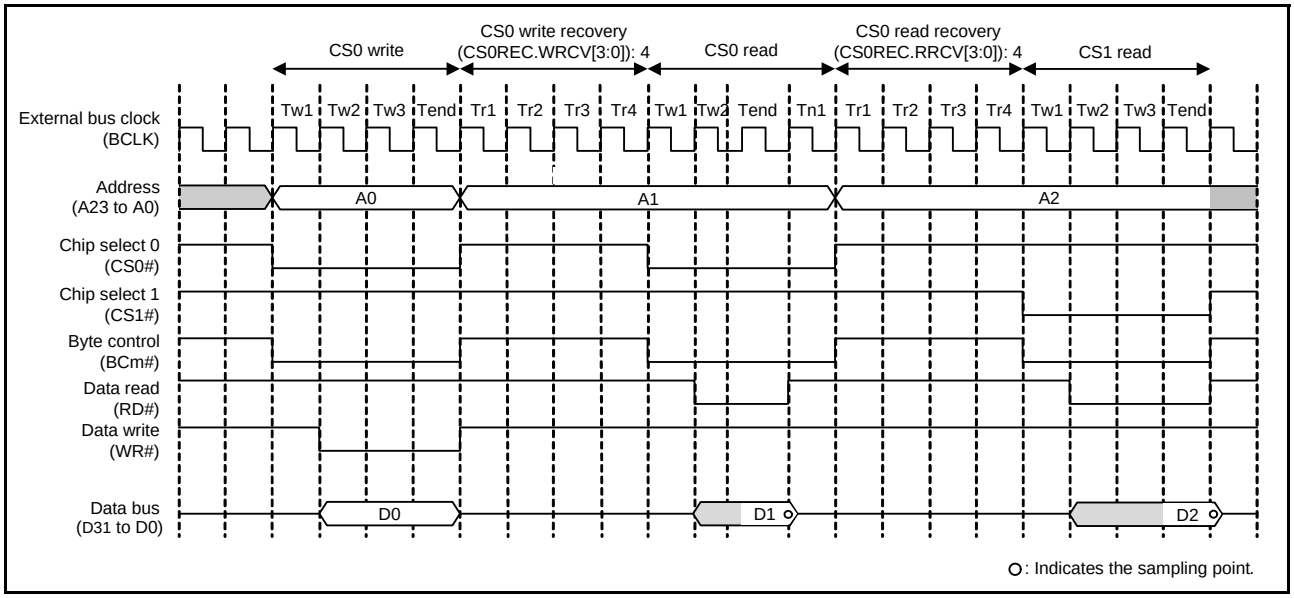


Figure 16.46 Example of Recovery Cycle Insertion (with Separate Bus Interface) (m = 0 to 3)

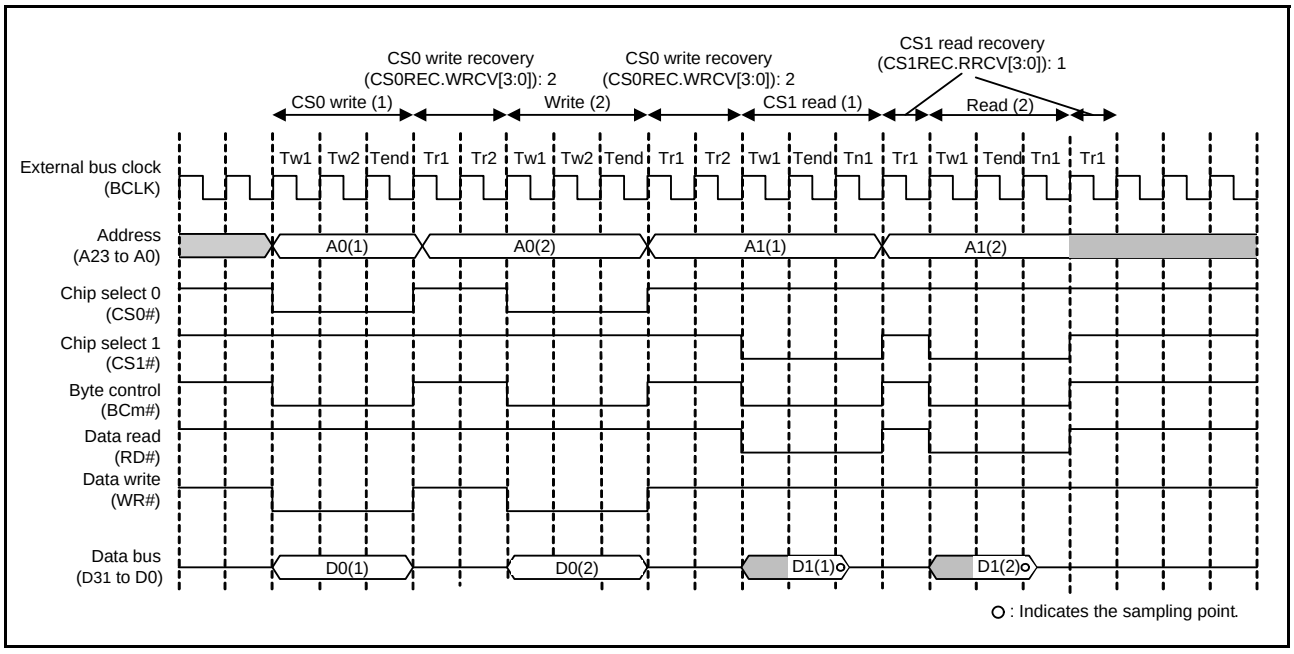


Figure 16.47 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Normal Access) (m = 0 to 3)

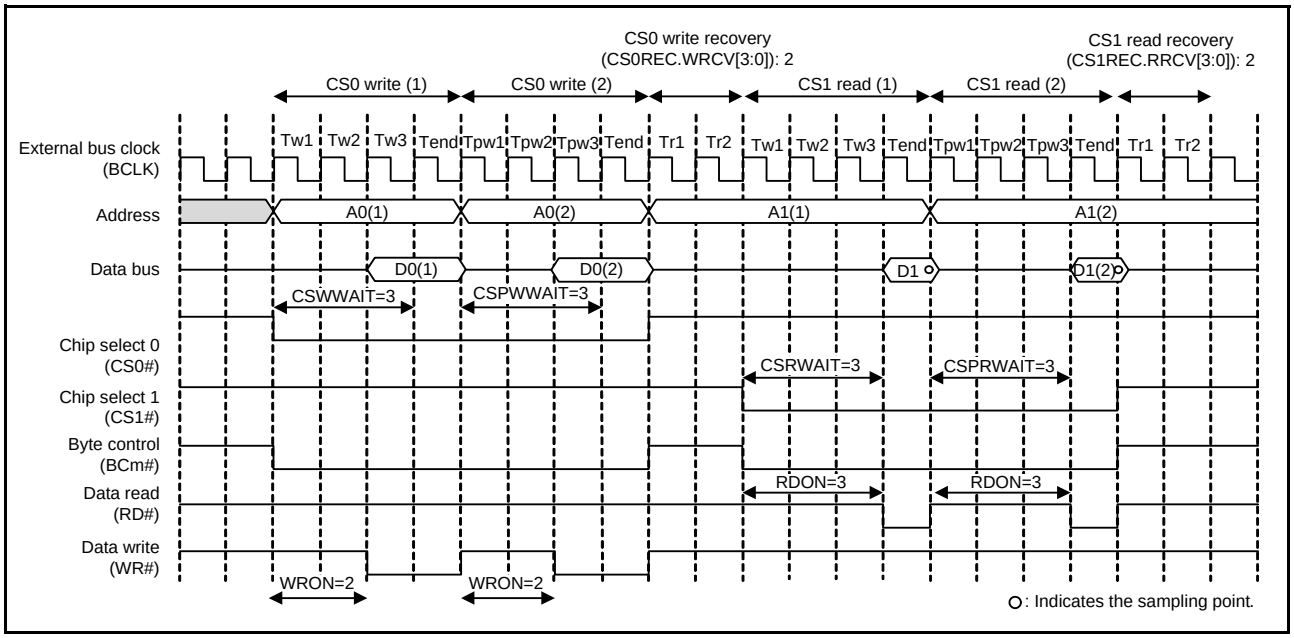


Figure 16.48 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Page Access) (m = 0 to 3)

Figure 16.49 and Figure 16.50 show examples of operations when the BCLK pin output selection bits are set for frequency-division of BCLK by 2.

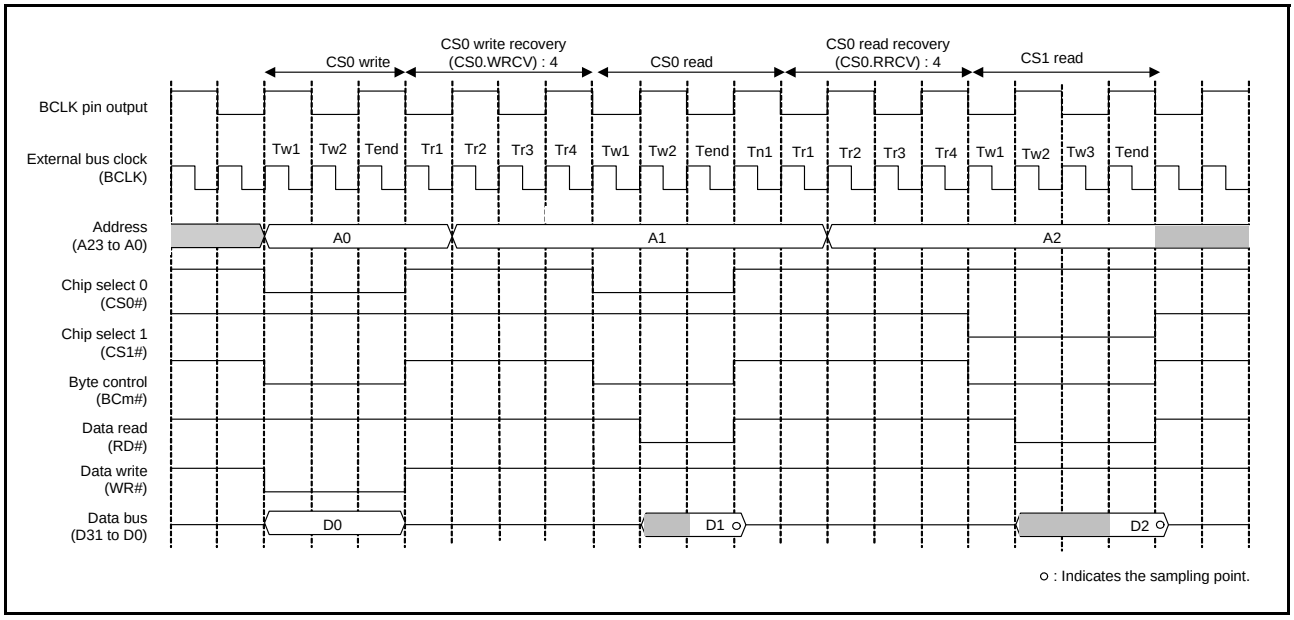


Figure 16.49 Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits Are Set for Frequency-Division of BCLK by 2 (For the Case of Normal Access Through a Separate Bus Interface; m = 0 to 3)

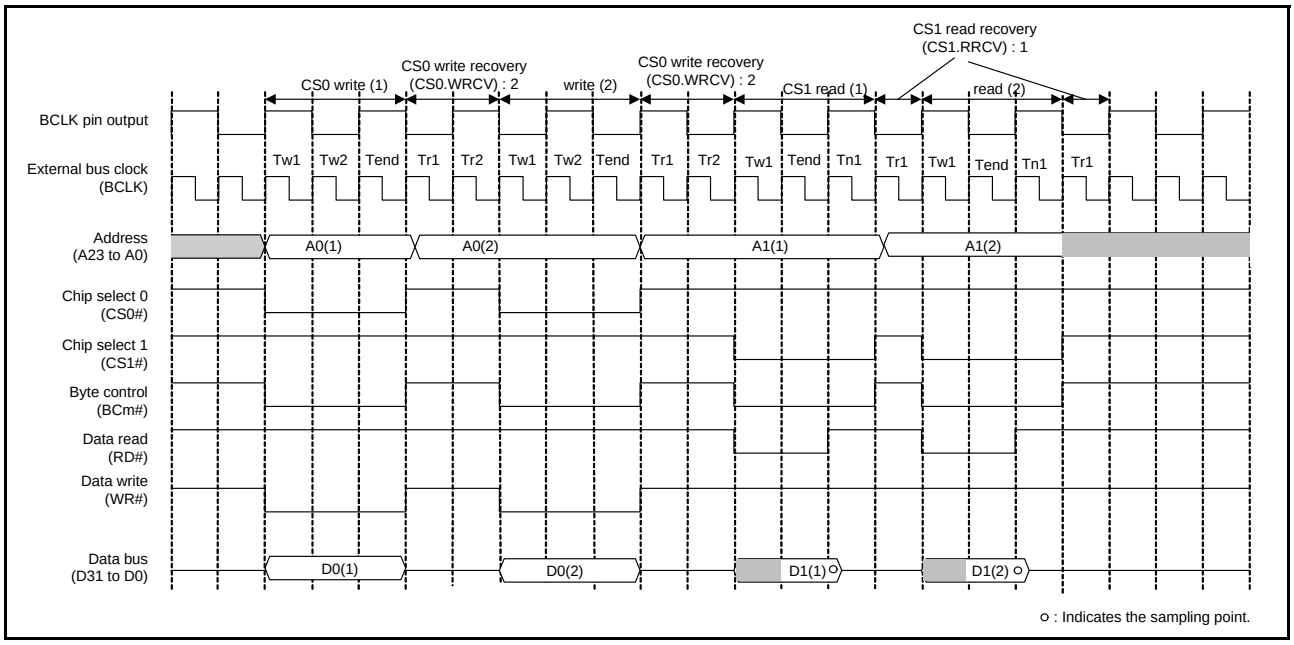


Figure 16.50 Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits Are Set for Frequency-Division of BCLK by 2 (For the Case where Bus Access is Divided up; m = 0 to 3)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 16.51 and Figure 16.52 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

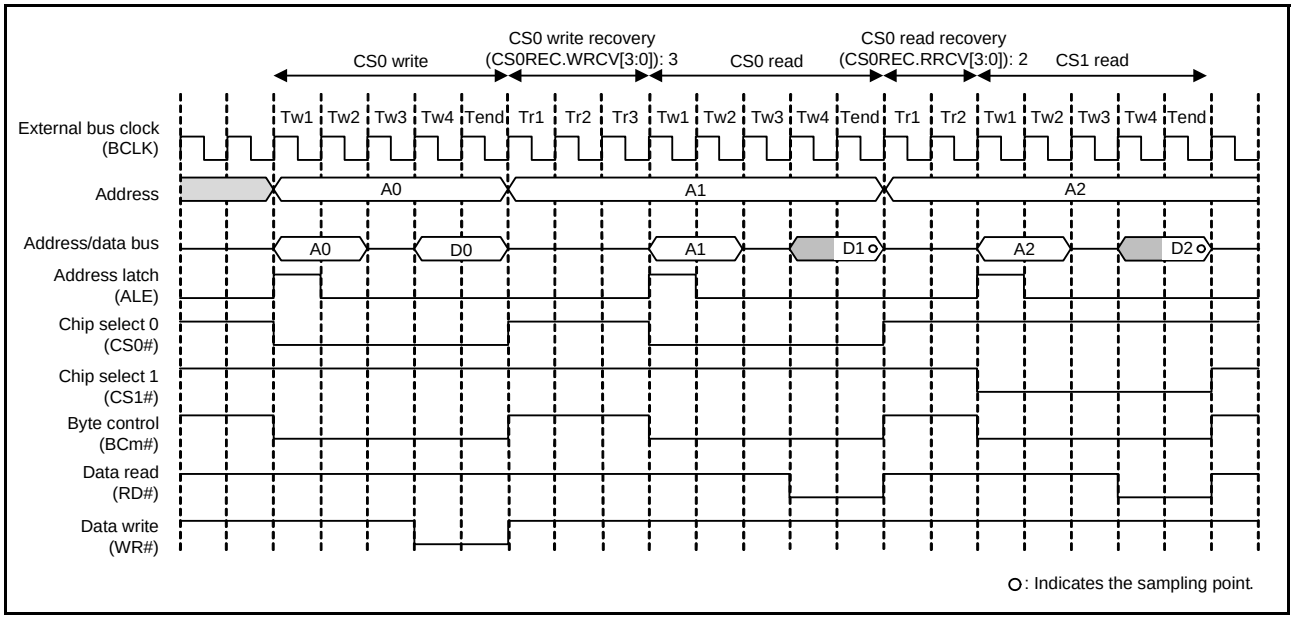


Figure 16.51 Example of Recovery Cycle Insertion (with Address/Data Multiplexed I/O Interface; m = 0, 1)

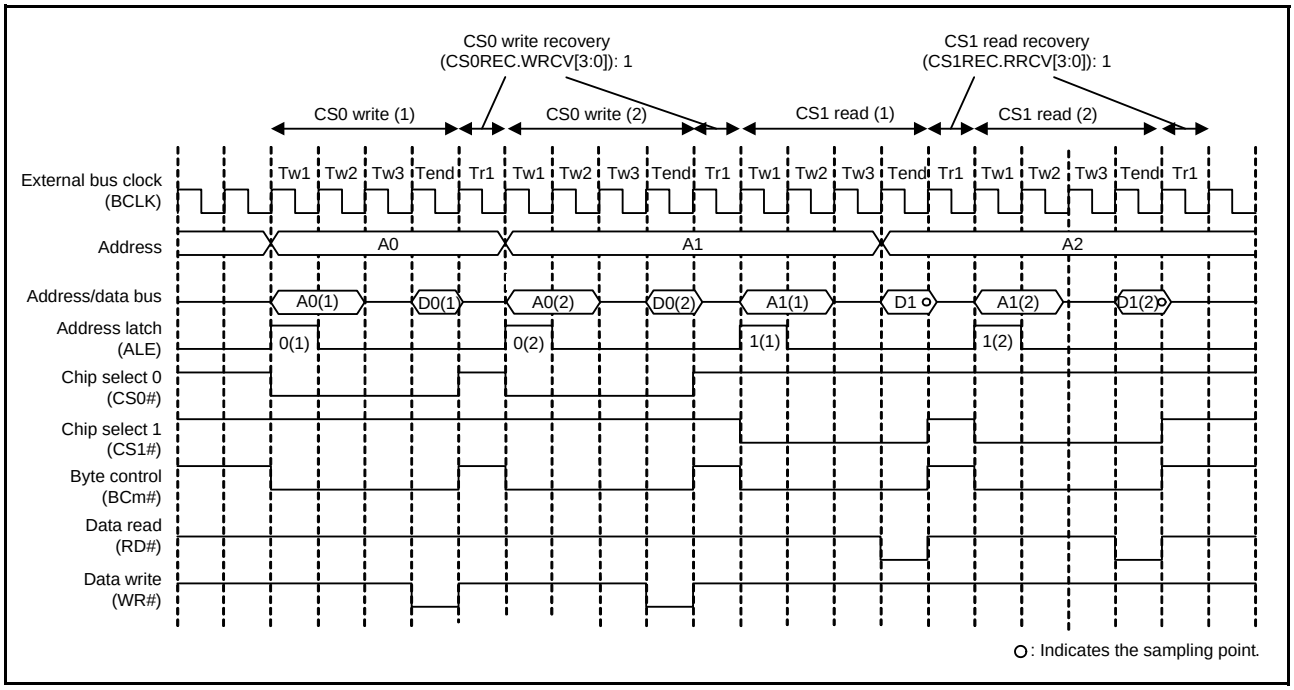


Figure 16.52 Example of Recovery Cycle Insertion When a Bus Access is Split (with Address/Data Multiplexed I/O Interface; m = 0, 1)

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

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 16.53 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.

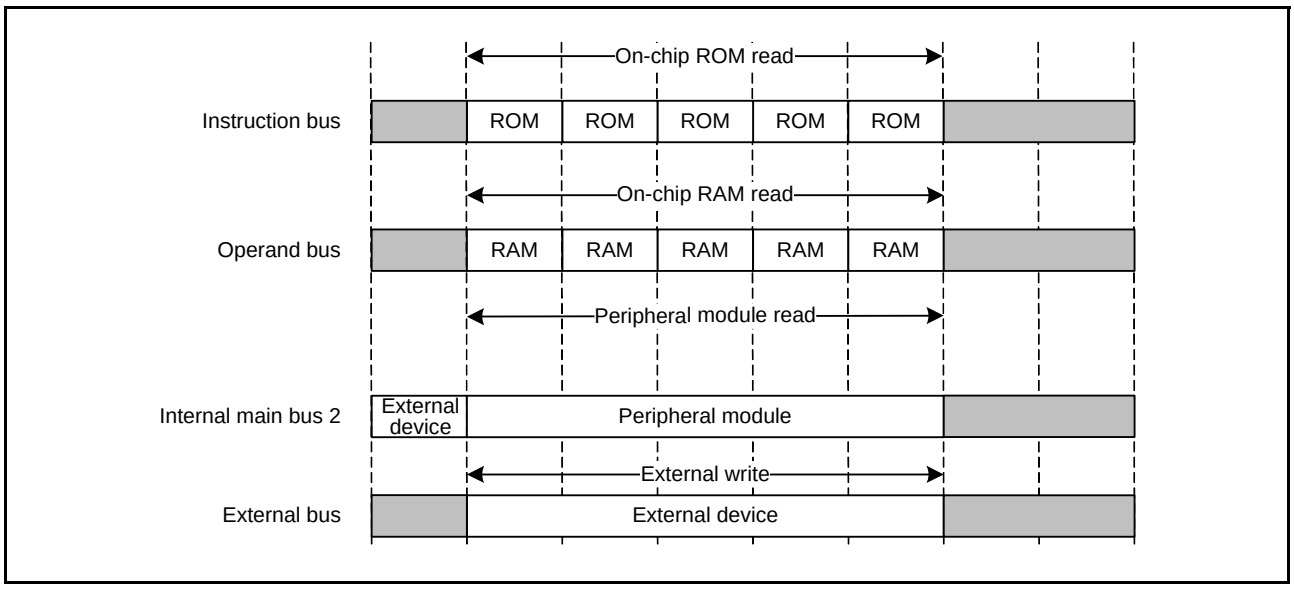


Figure 16.53 Example of Operation when the Write-Buffer Function is in Use

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 is 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).

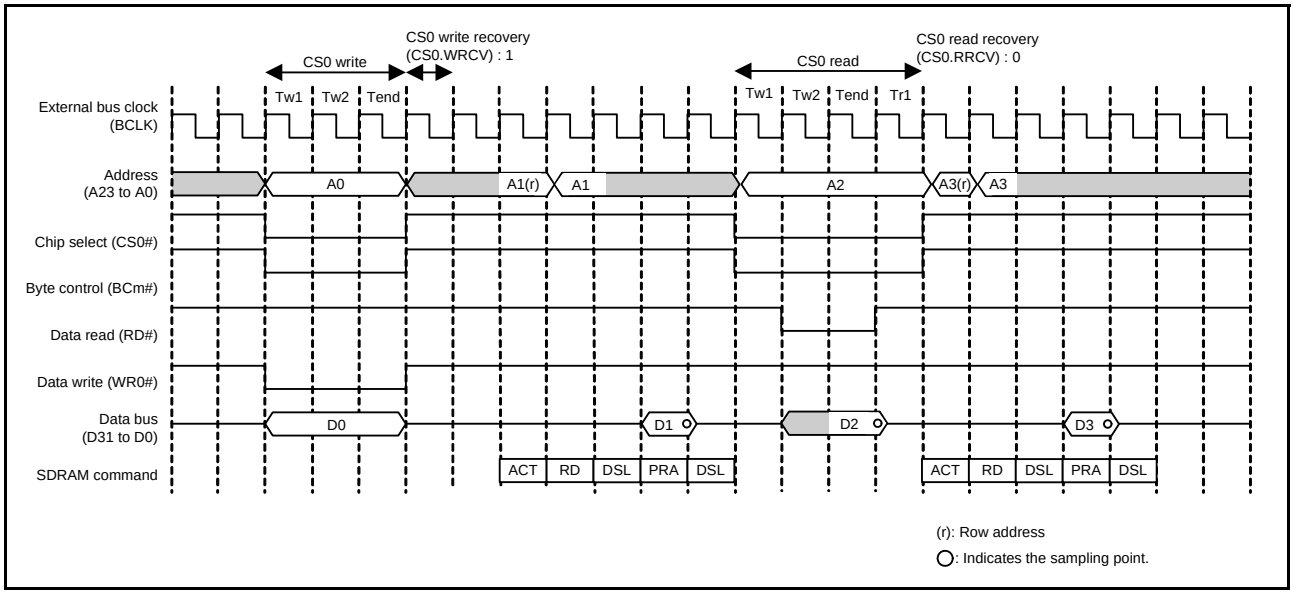


Figure 16.54 Example of Recovery Timing (for SDRAM Access)

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	• 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 cleared 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 16.55 and Figure 16.56 show timing examples of transition to self-refresh mode and recovery from self-refresh mode, respectively.

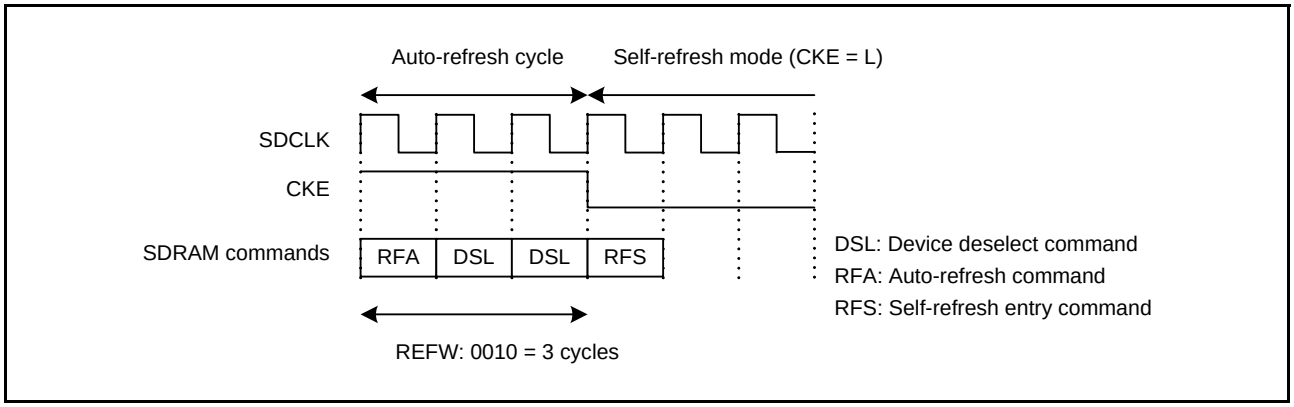


Figure 16.55 Timing Example of Transition to Self-Refresh Mode (when SDRFCR.REFW[3:0] = 0010b: 3 Cycles)

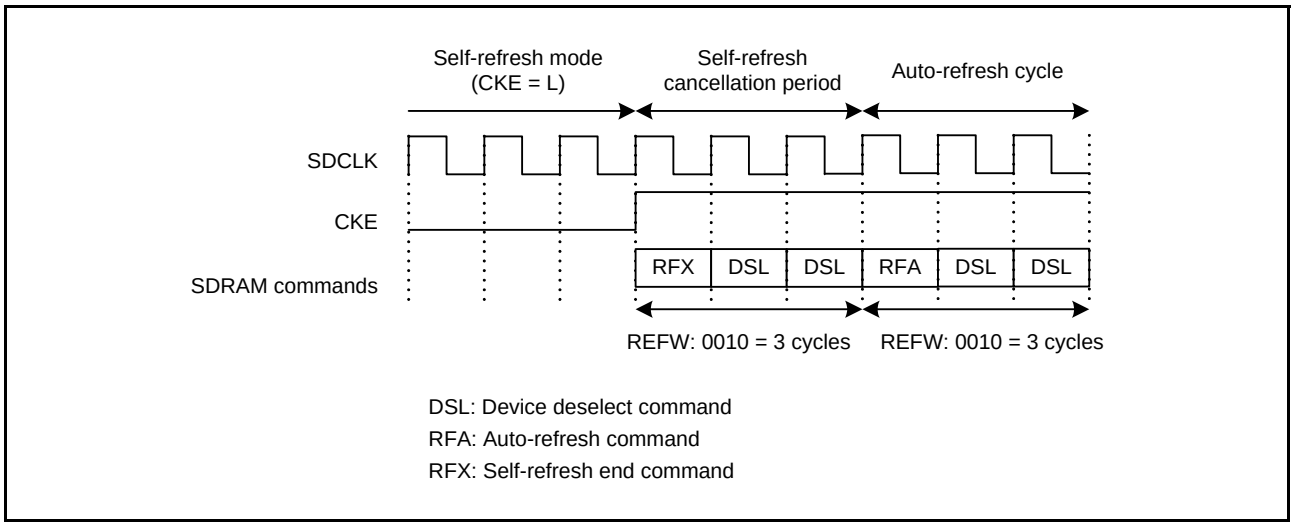


Figure 16.56 Timing Example of Recovery from Self-Refresh Mode

(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 Modee, 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 16.57 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 cleared 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) Clear the IOKEEP bit in DPSBYCR to 0 to release the I/O ports from the held state.
6. Clear 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 cleared to 0 and set the SFEN bit in SDSELF to 0 to cancel self-refresh mode.

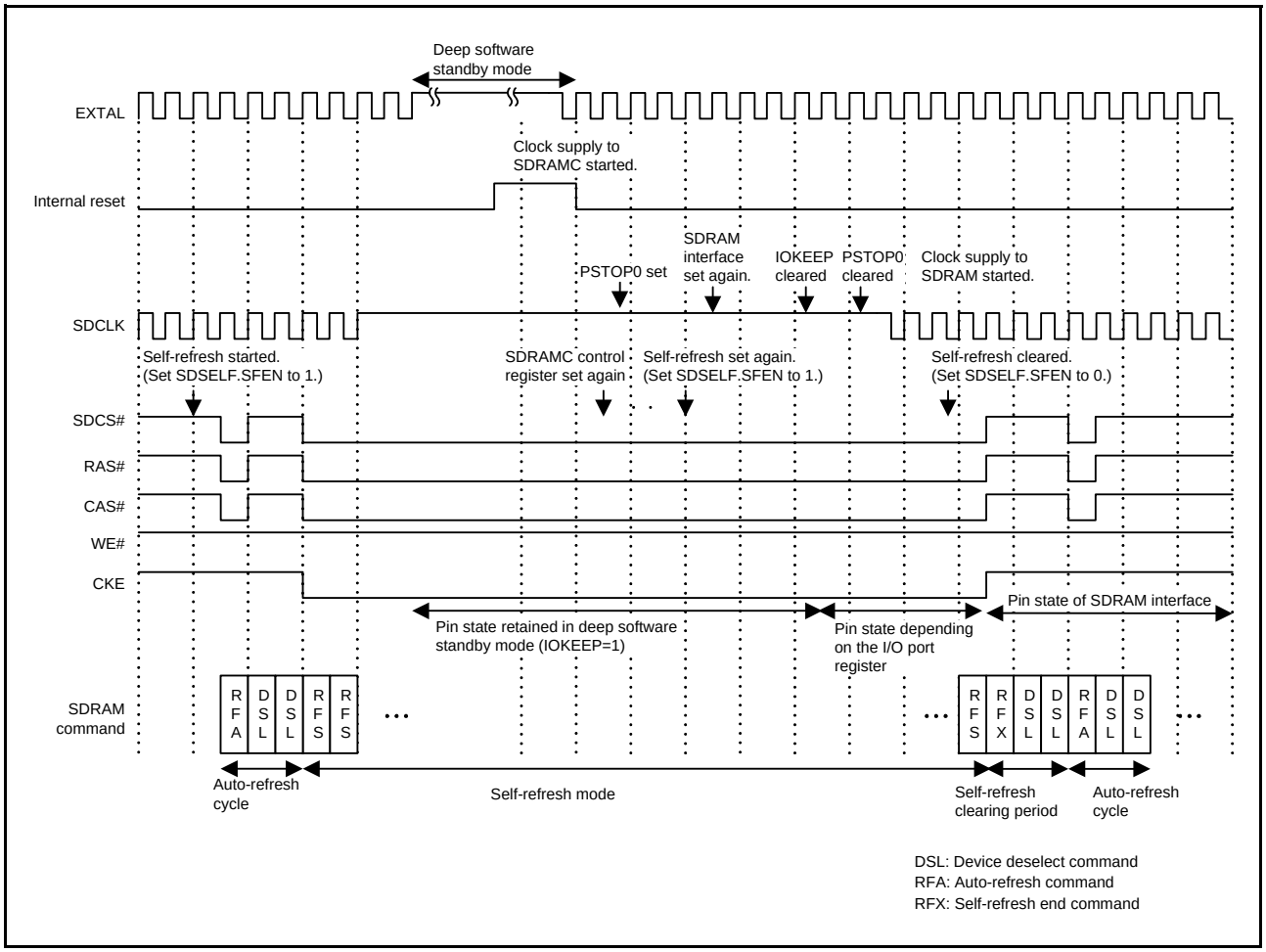


Figure 16.57 Timing Example of Self-Refresh Cycle (in Deep Software Standby Mode)

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 16.58 shows an example of the timing of an auto-refresh cycle.

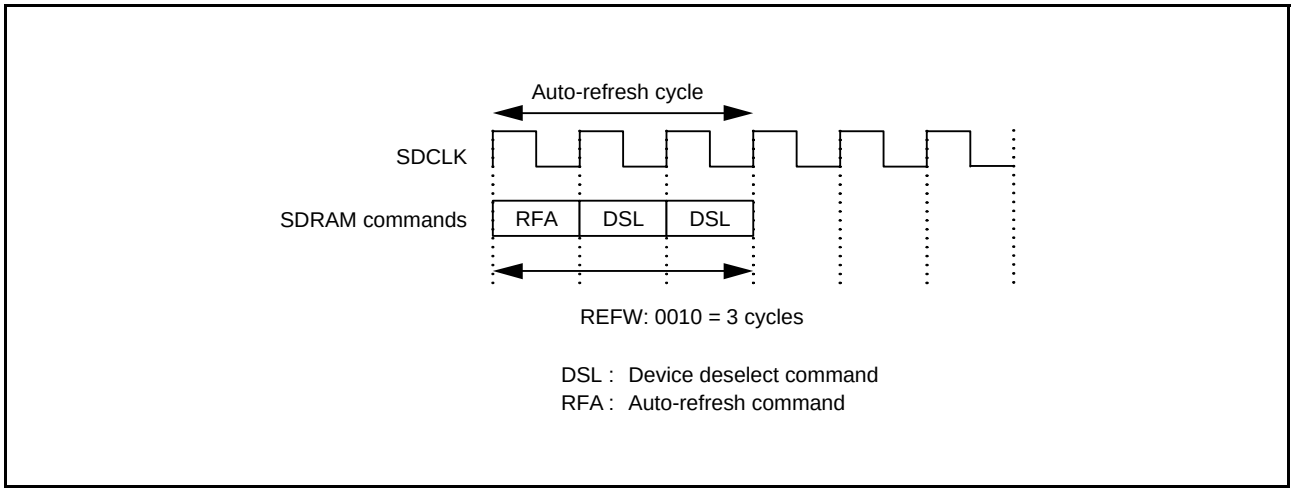


Figure 16.58 Timing Example of Auto-Refresh Cycle (1)

Figure 16.59 and Figure 16.60 show examples of operation when an auto-refresh request is generated during single access and continuous access, respectively.

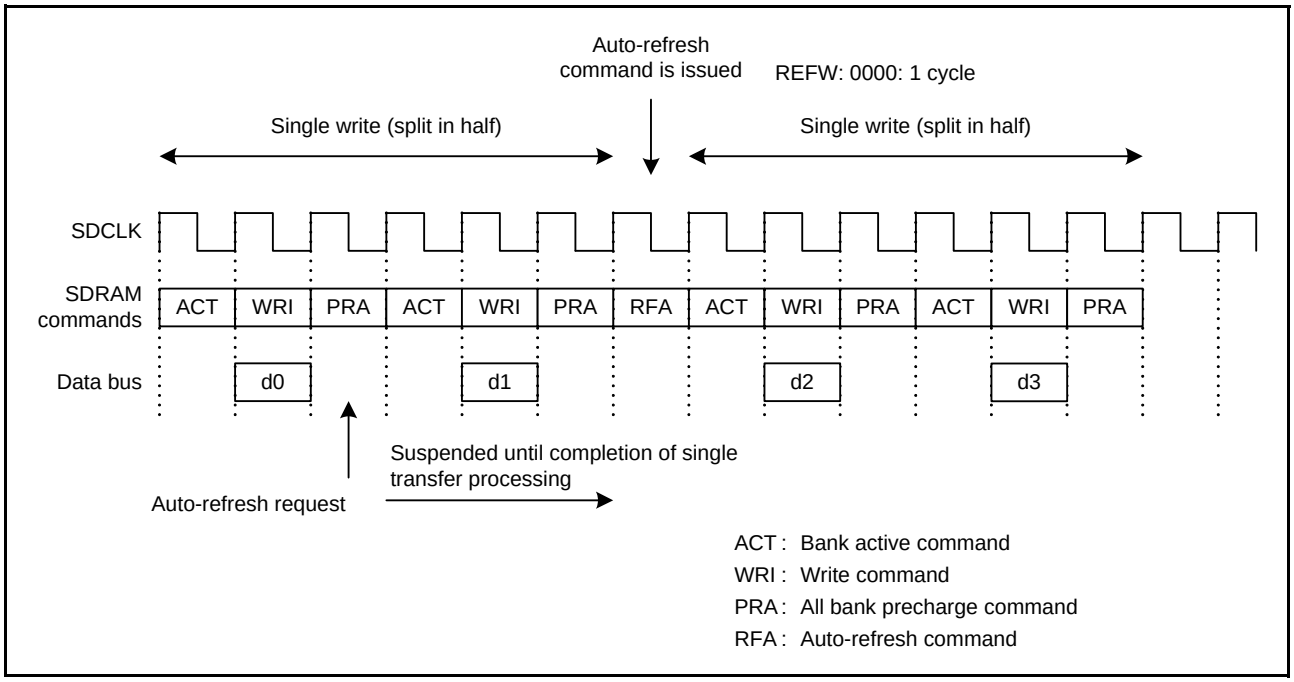


Figure 16.59 Timing Example of Auto-Refresh Cycle (2)
(Auto-Refresh Request is Made during Single Access)

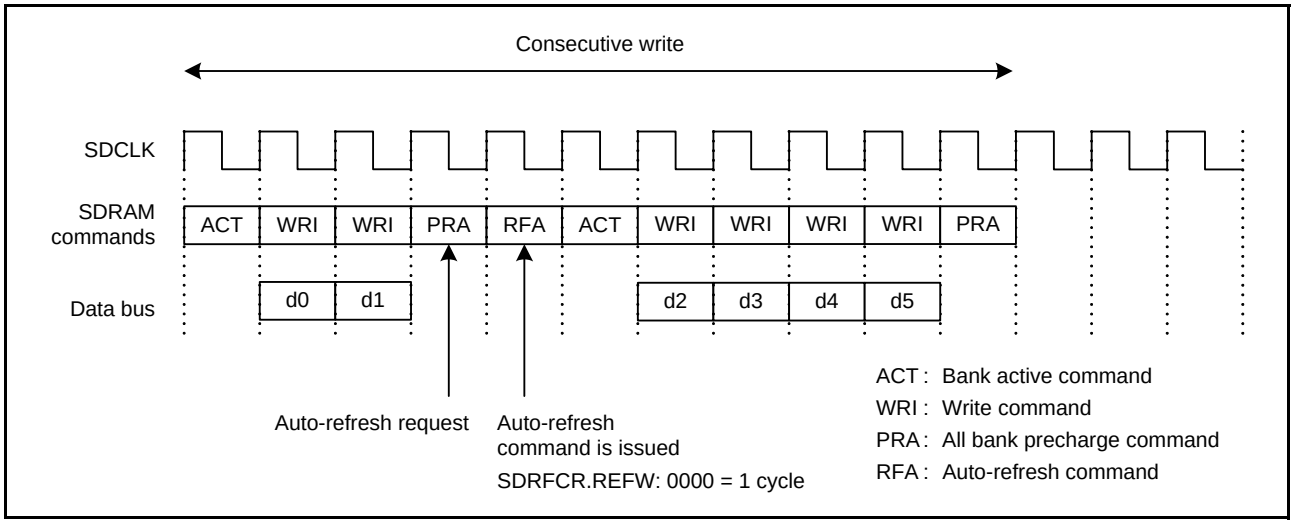


Figure 16.60 Timing Example of Auto-Refresh Cycle (3)
(Auto-Refresh Request is Made during Continuous Access)

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 16.61 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.

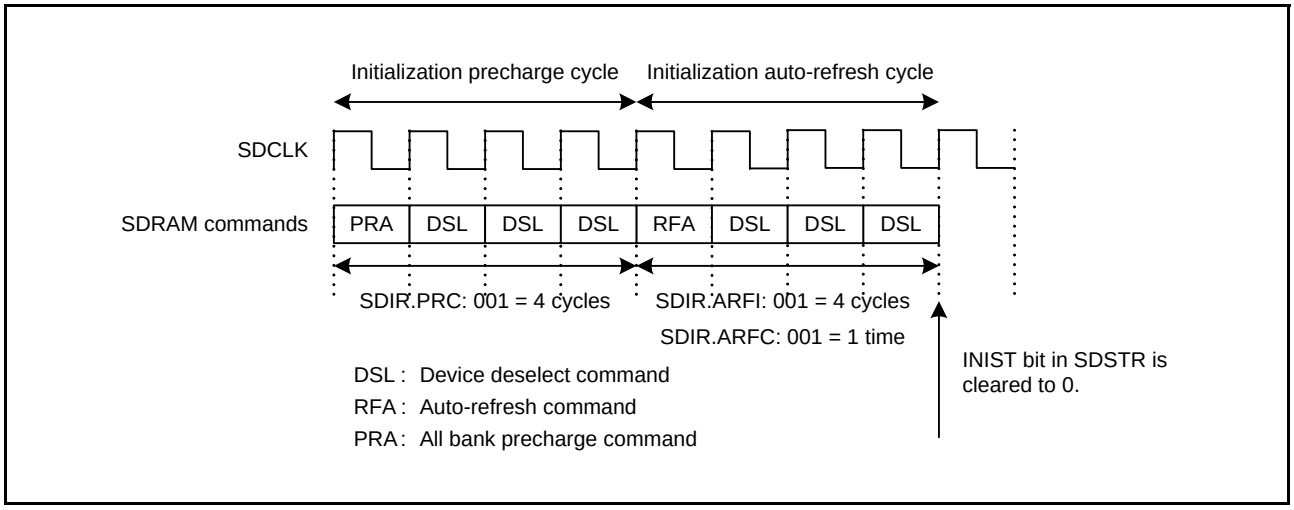


Figure 16.61 Timing Example of SDRAM Initialization Sequence

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 16.62 and Figure 16.63 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.

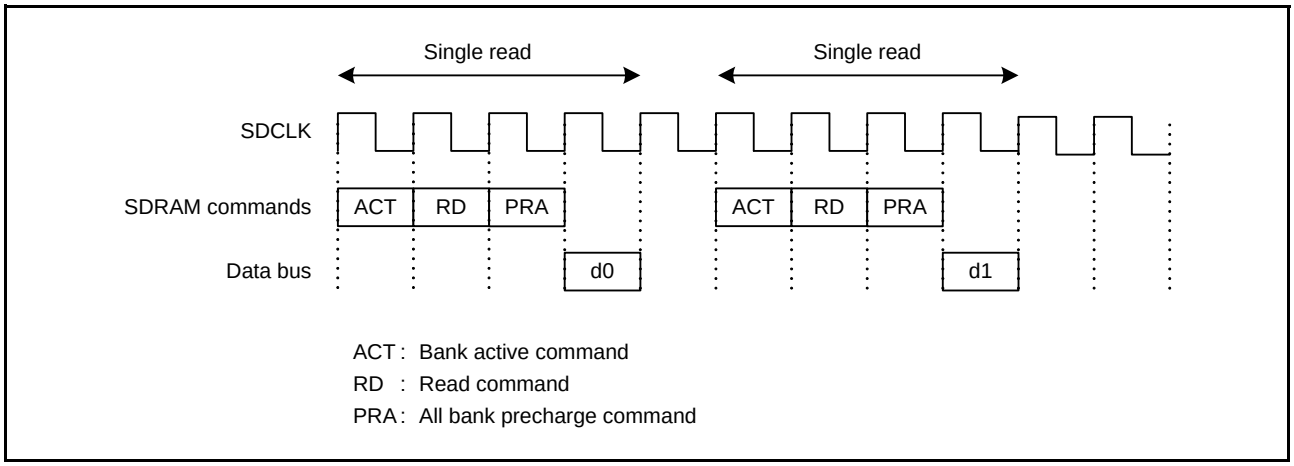


Figure 16.62 Timing Example of Single Read (SDTR.CL[2:0] = 010b: 2 Cycles)

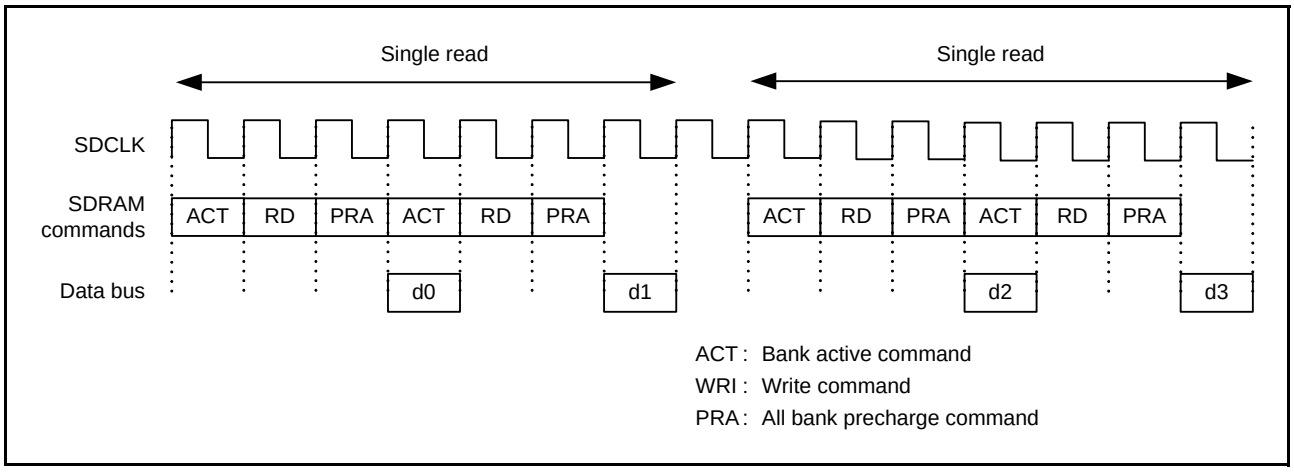


Figure 16.63 Timing Example of Single Read (Cluster Transfer by EXDMAC or Block Transfer in Single Address Mode with SDAMOD.BE = 0 and SDTR.CL[2:0] = 010b: 2 Cycles)

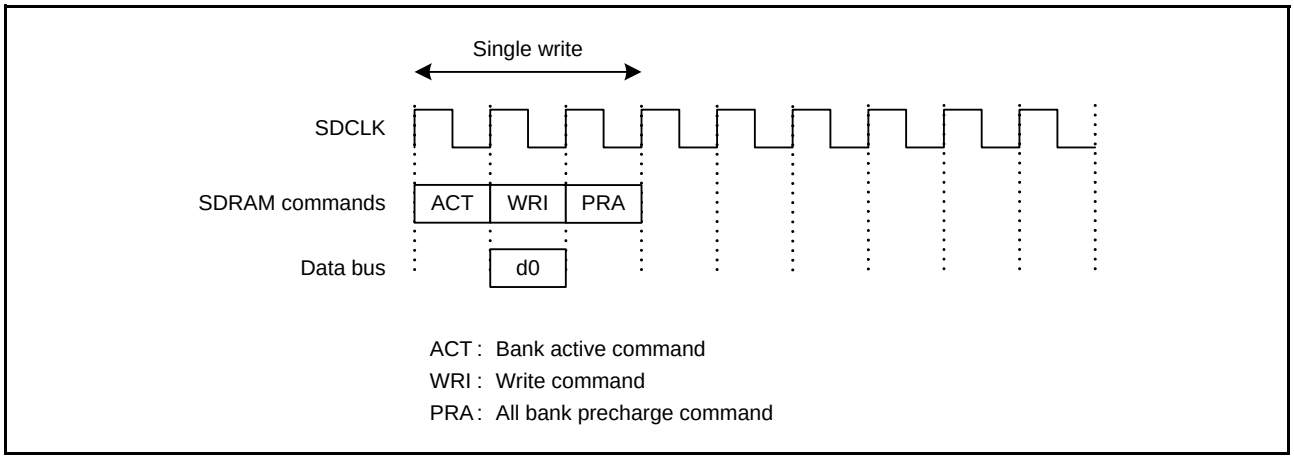


Figure 16.64 Timing Example of Single Write (when the Shortest Timing is Set)

(2) Consecutive Access

Figure 16.65 and Figure 16.66 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 16.67 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.

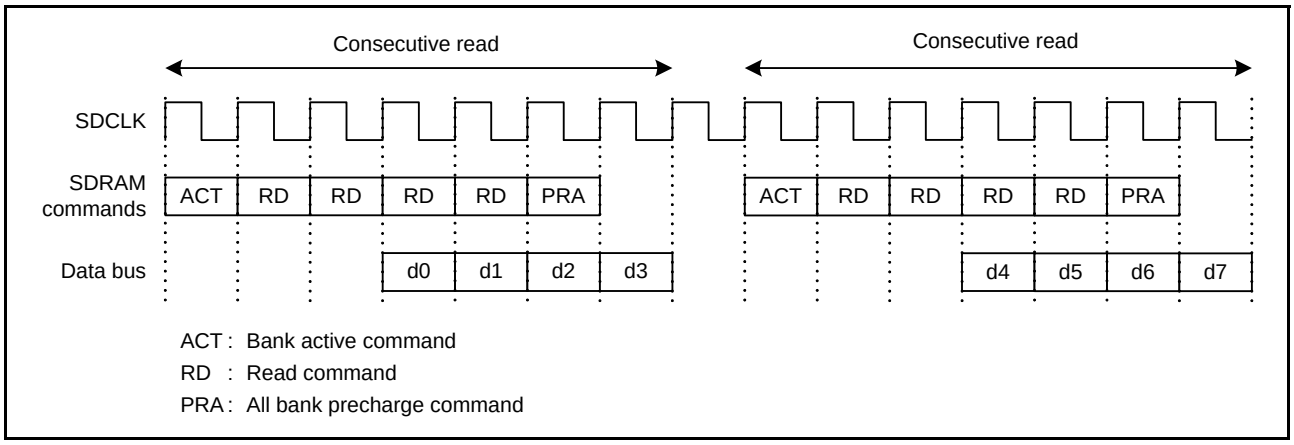


Figure 16.65 Timing Example of Consecutive Read (SDAMOD.BE = 1 and SDTR.CL[2:0] = 010b: 2 Cycles)

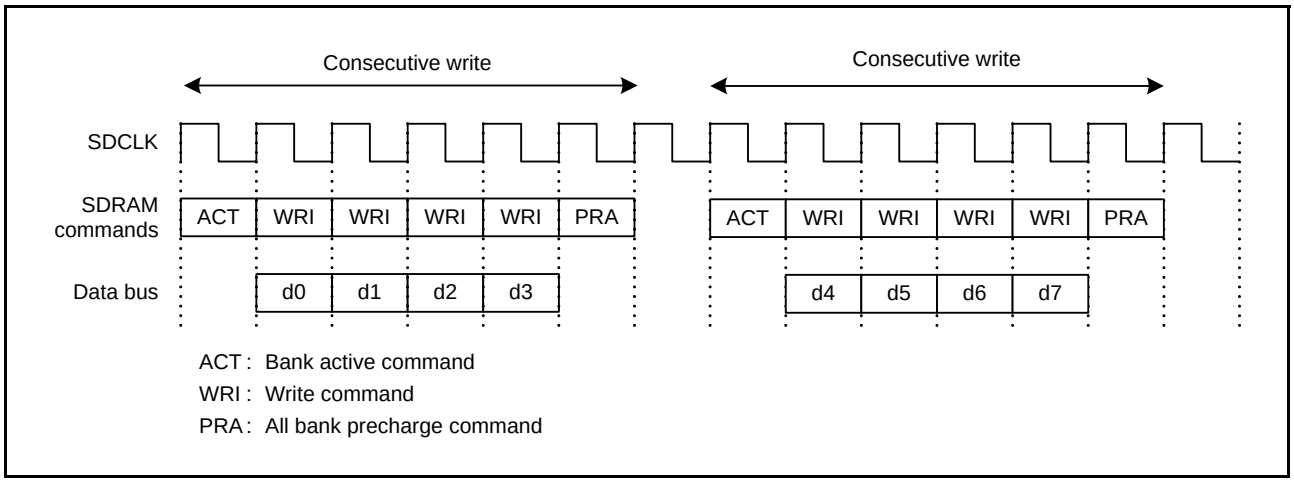


Figure 16.66 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set)

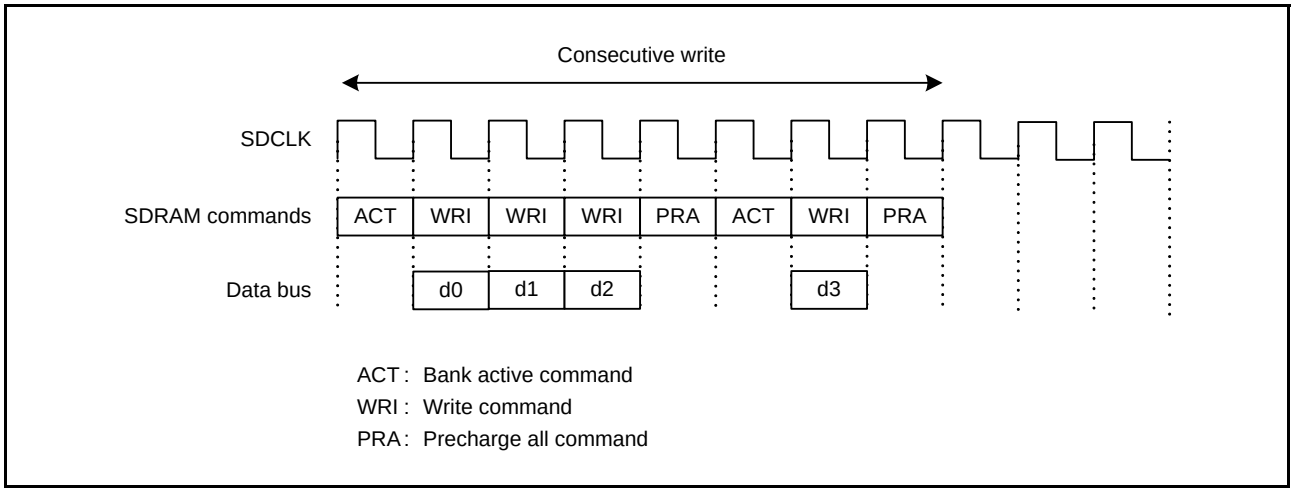


Figure 16.67 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set) with More Than One Row Address Accessed)

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 16.68 shows the mode register setting timing.

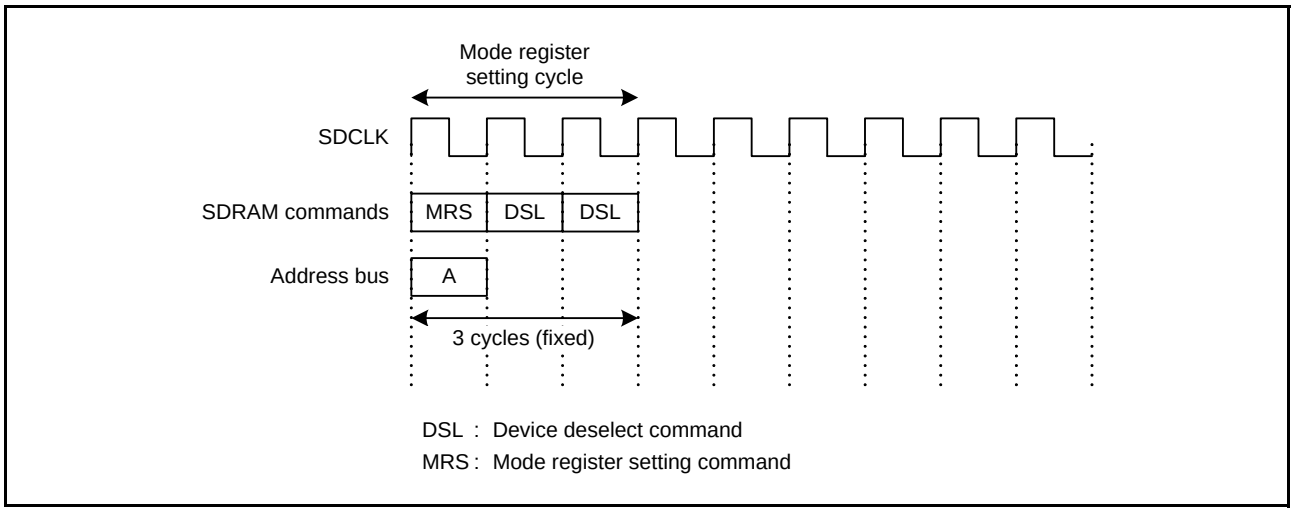


Figure 16.68 Mode Register Setting Timing

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 16.69 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.

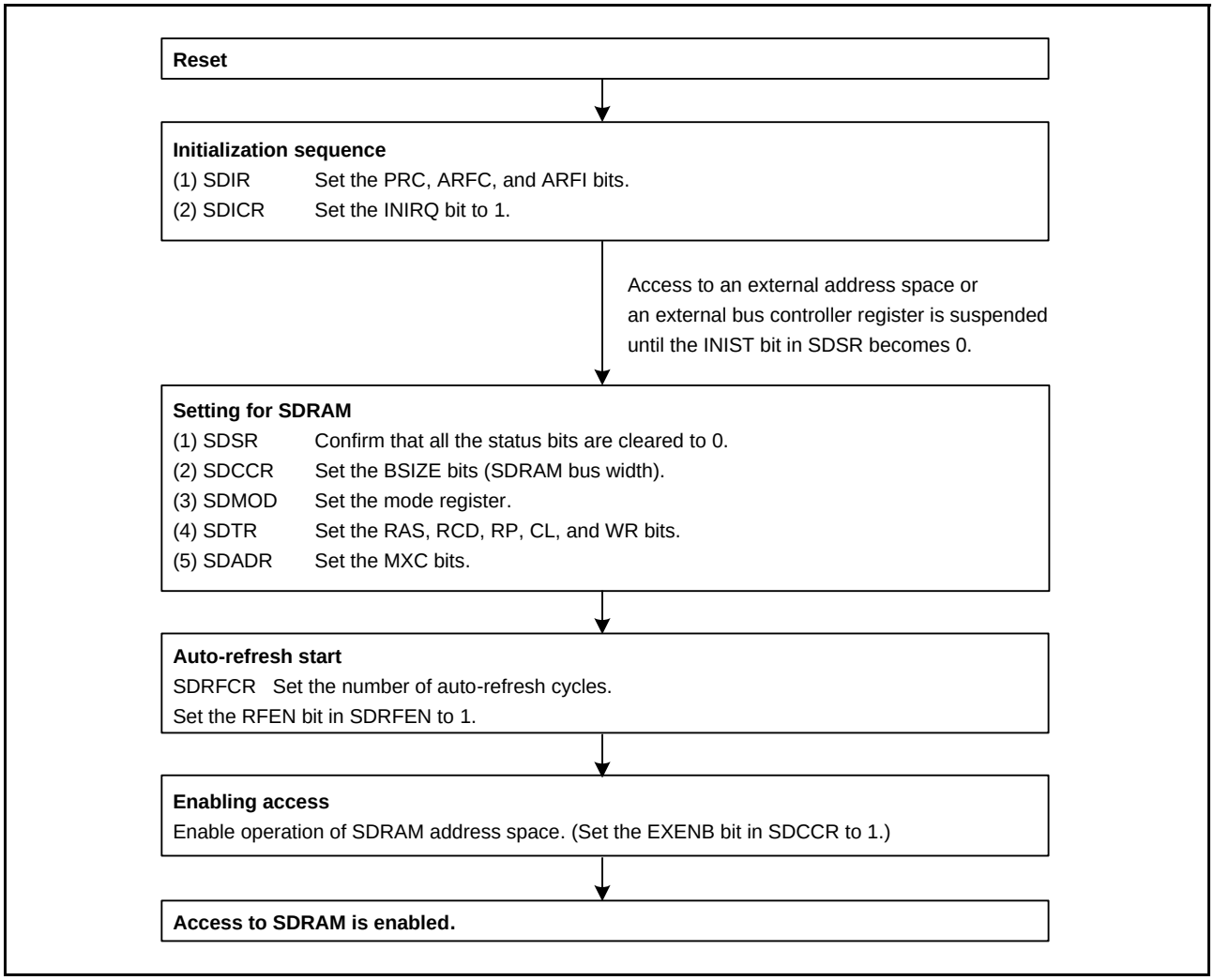


Figure 16.69 SDRAMC Setting Procedure

16.6.12.2 Procedure for Transition to and Recovery from Self-Refresh Mode

Figure 16.70 shows the procedure for transition to and recovery from self-refresh mode.

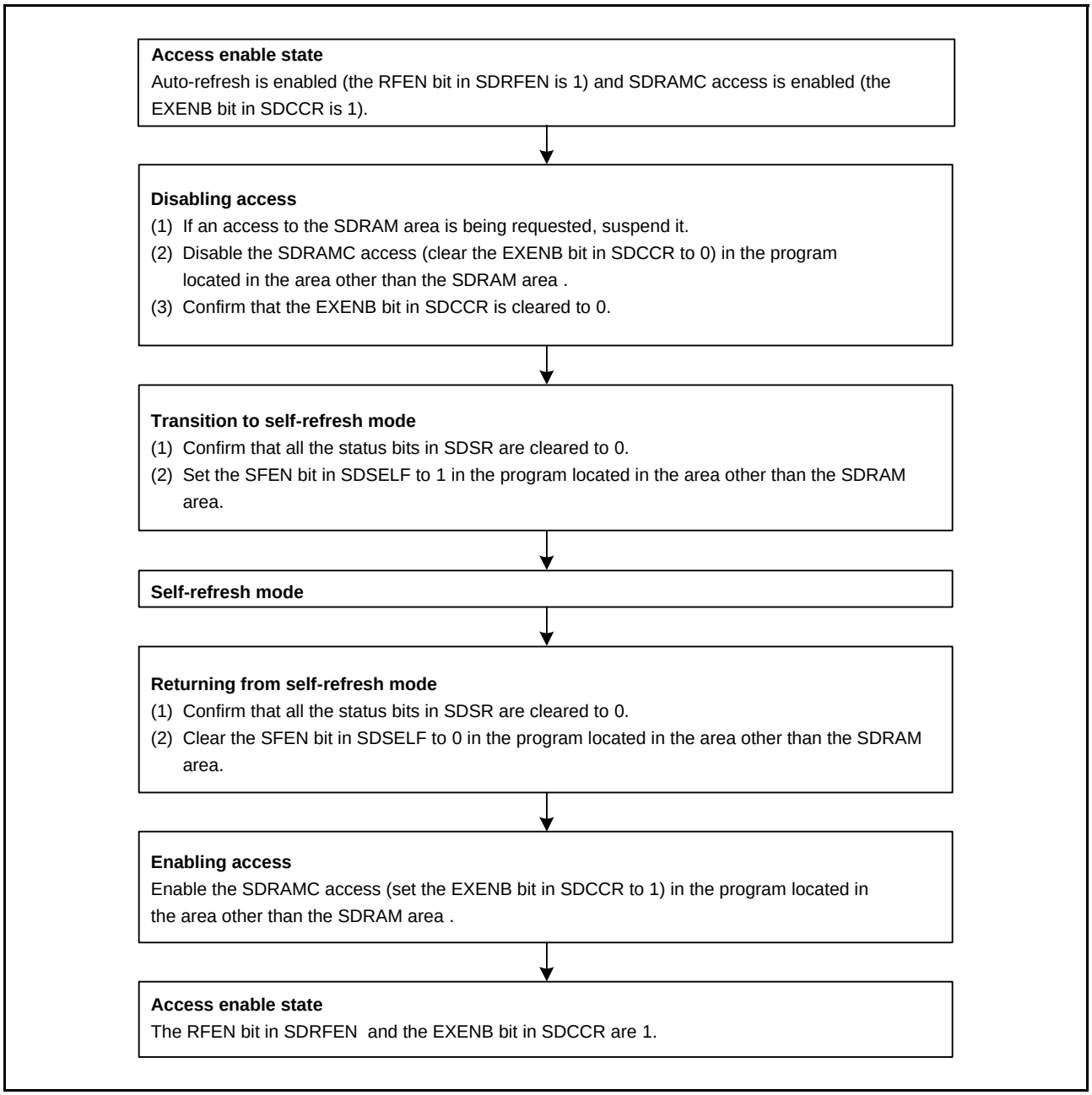


Figure 16.70 Procedure for Transition to and Recovery from Self-Refresh Mode

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 16.71 shows the procedure for transition to and recovery from self-refresh mode in deep software standby mode.

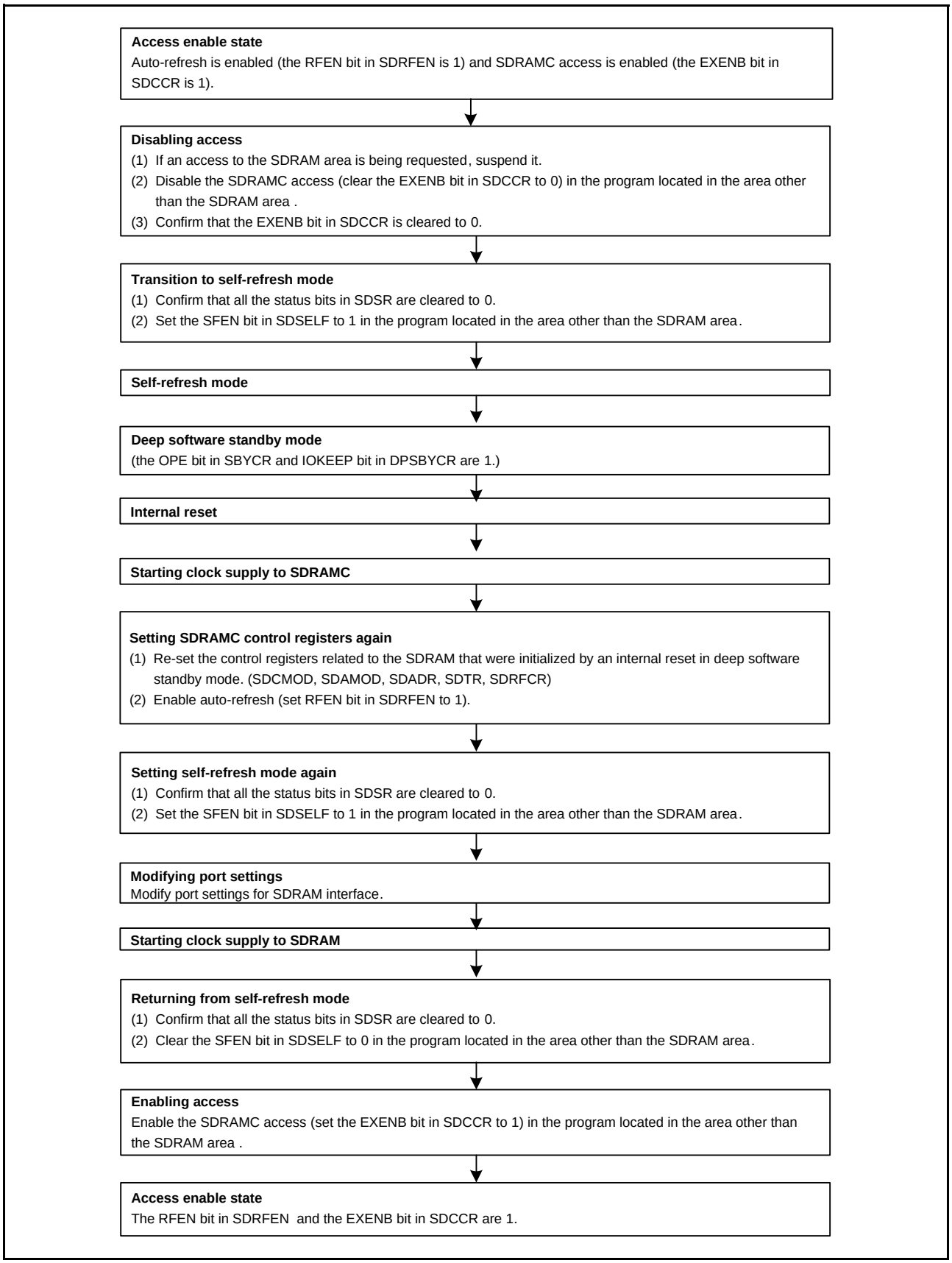


Figure 16.71 Procedure for Transition to and Recovery from Self-Refresh Mode in Deep Software Standby Mode

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 16.72 to Figure 16.76 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 16.76.

Table 16.16 Correspondence between Target Figures and STDR 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 16.72	010	3	00	1	001	2	010	2
Figure 16.73	000	1	01	2	001	2	010	2
Figure 16.74	000	1	01	2	001	2	011	3
Figure 16.75 Figure 16.76	010	3	00	1	000	1	010	2

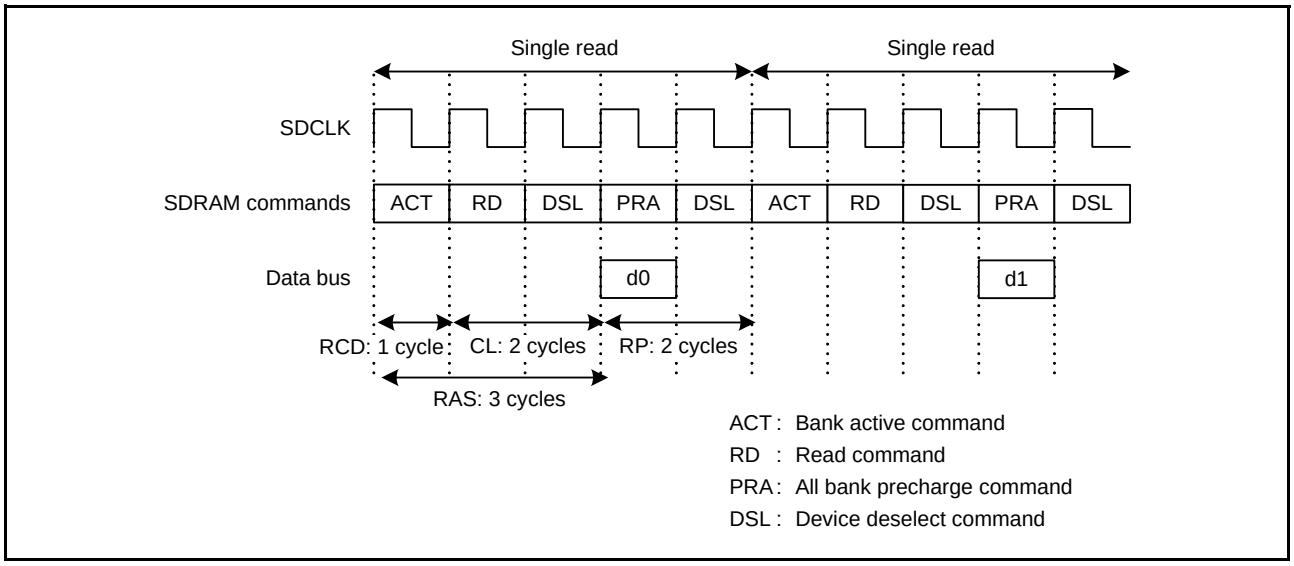


Figure 16.72 Timing Example of Single Read (1)

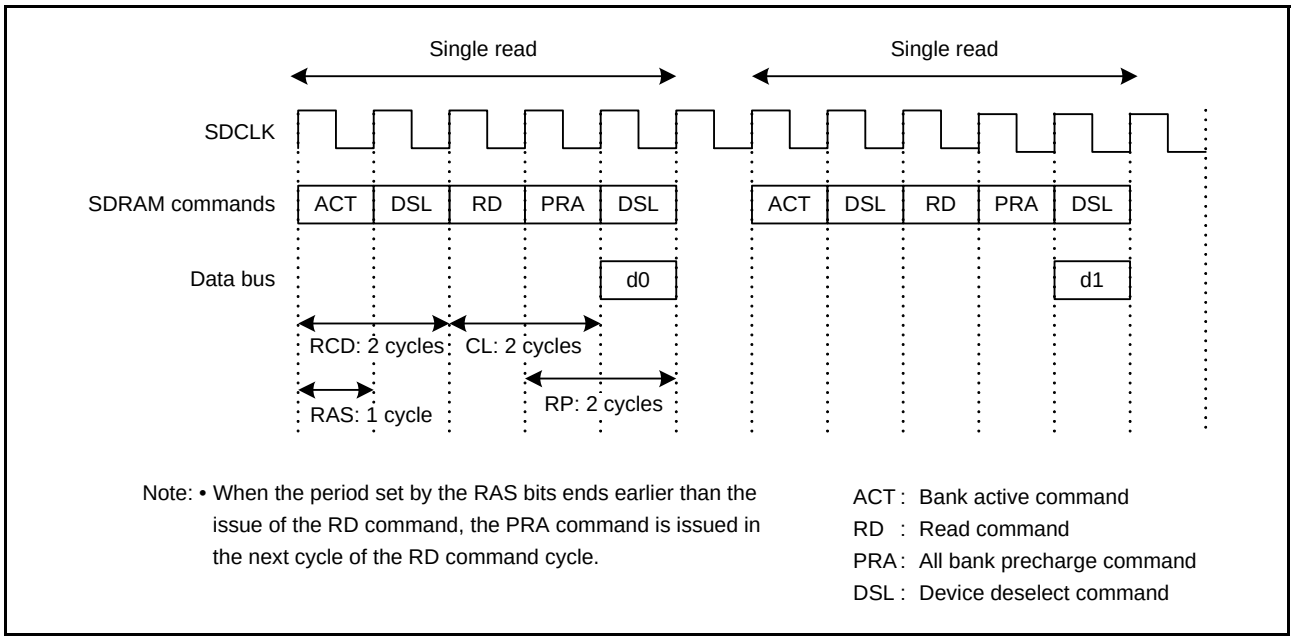


Figure 16.73 Timing Example of Single Read (2)

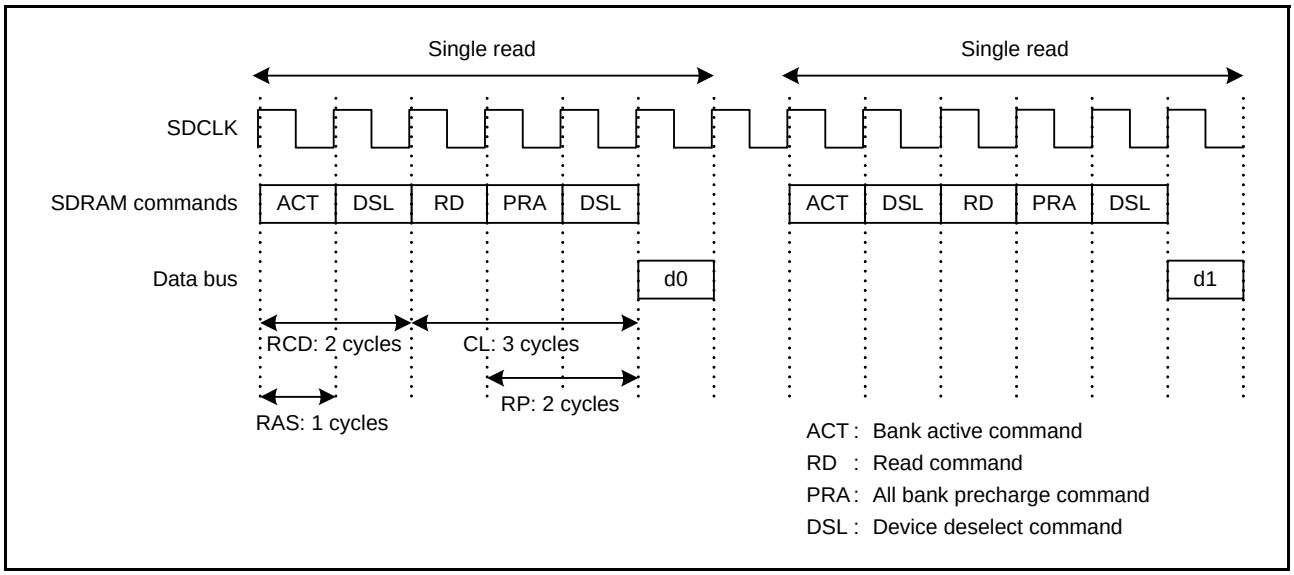


Figure 16.74 Timing Example of Single Read (3)

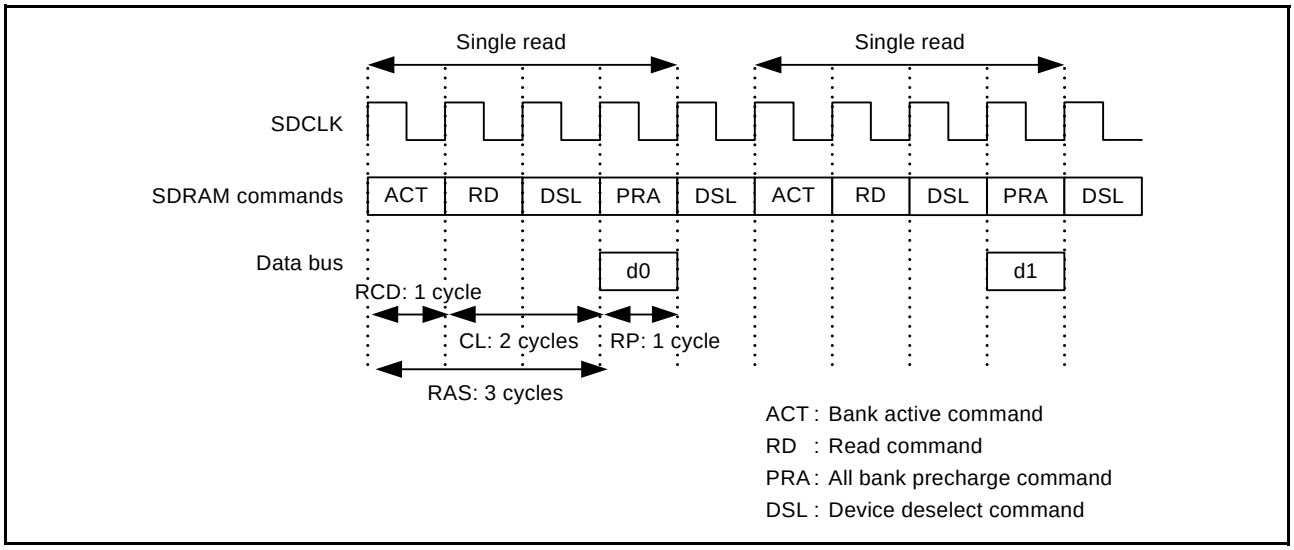


Figure 16.75 Timing Example of Single Read (4)

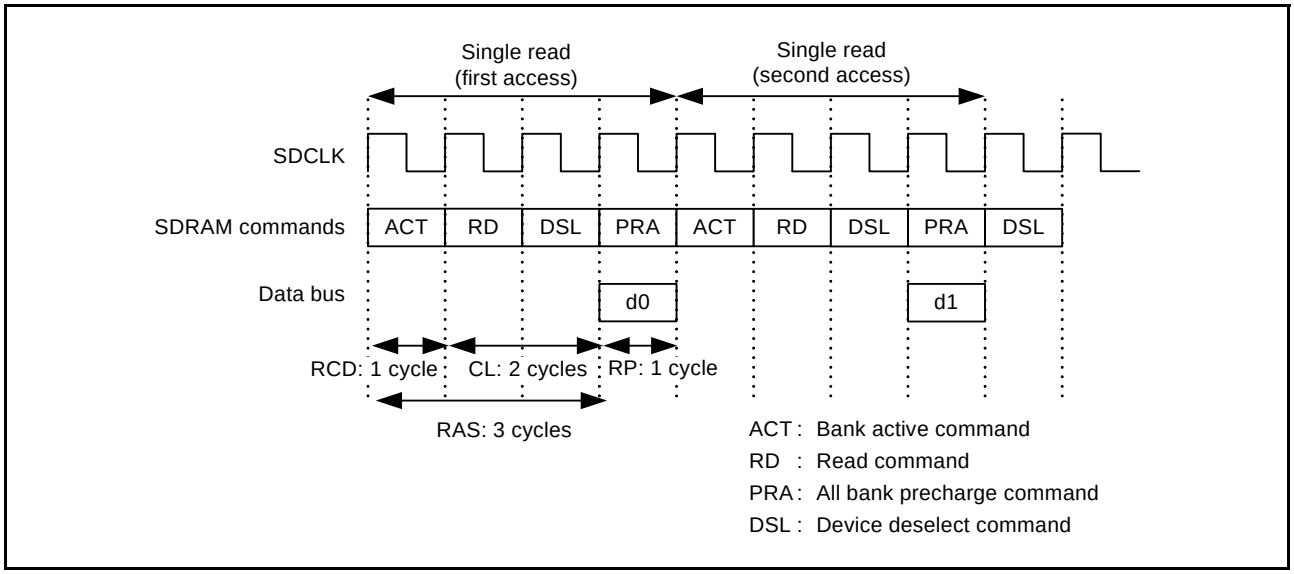


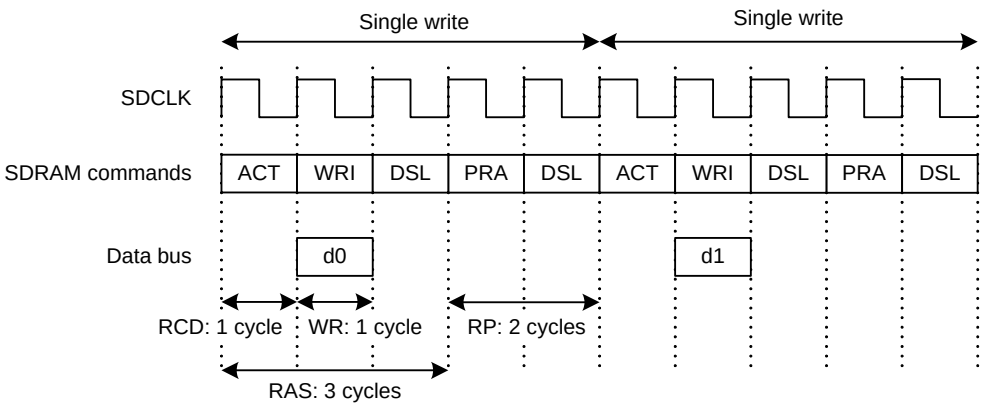
Figure 16.76 Timing Example of Single Read (5) (Two Bus Accesses Occur for One Transfer Request.)

(2) Single Write Timing Examples

Figure 16.77 to Figure 16.81 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 16.81.

Table 16.17 Correspondence between Target Figures and STDR Register Settings (Single Write Timing)

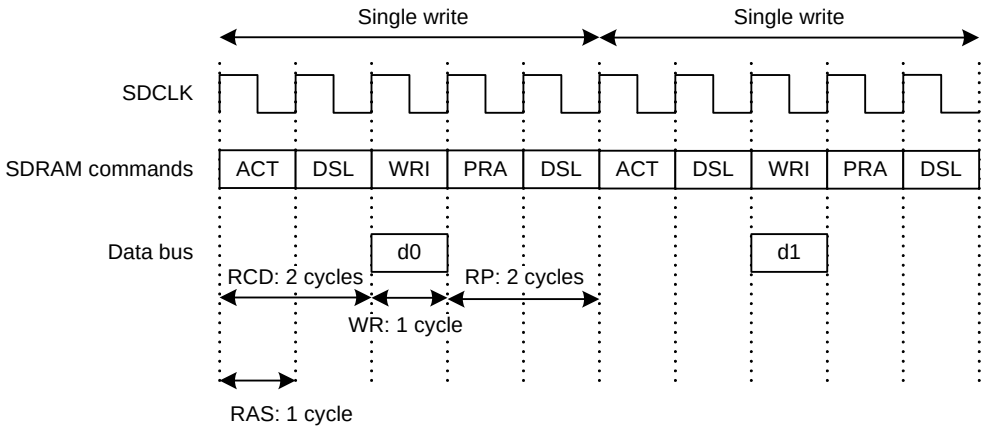
Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 16.77	010	3	00	1	001	2	0	1
Figure 16.78	000	1	01	2	001	2	0	1
Figure 16.79	000	1	01	2	001	2	1	2
Figure 16.80 Figure 16.81	010	3	00	0	000	2	0	1



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

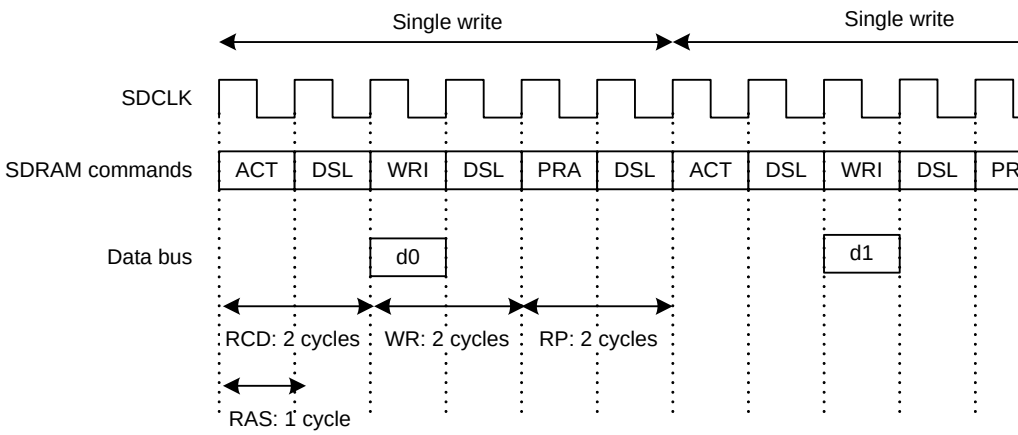
Figure 16.77 Timing Example of Single Write (1)



Note: • When the period specified by the RAS bits is shorter than the period specified by WR bits starting from the issue of the WRI command, the period specified by the WR bits is applied.

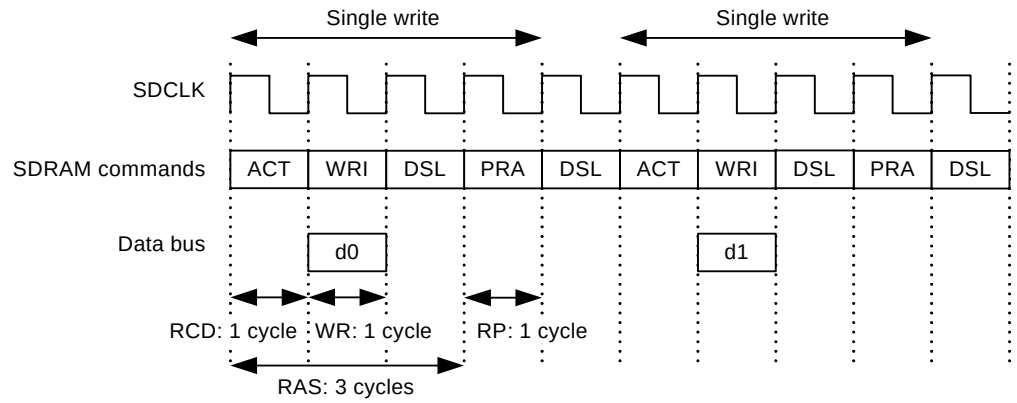
ACT : Bank active command
WRI : Write command
PRA : All bank precharge command
DSL : Device deselect command

Figure 16.78 Timing Example of Single Write (2)



ACT : Bank active command
WRI : Write command
PRA : All bank precharge command
DSL : Device deselect command

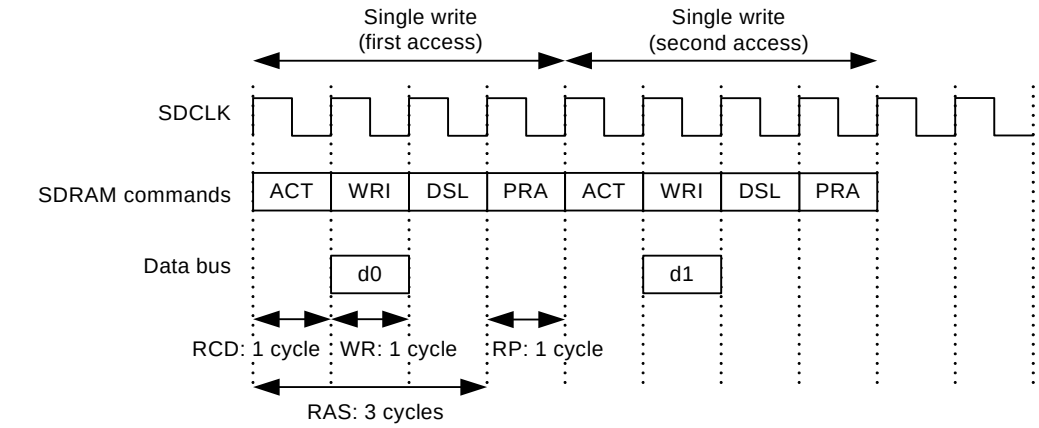
Figure 16.79 Timing Example of Single Write (3)



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 16.80 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 16.81 Timing Example of Single Write (5) (Two Bus Accesses Occur for One Transfer Request)

(3) Consecutive Read Timing Examples

Figure 16.82 to Figure 16.84 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 STDR 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 16.82	010	3	00	1	001	2	010	2
Figure 16.83	000	1	01	2	001	2	010	2
Figure 16.84	000	1	01	2	001	2	011	3

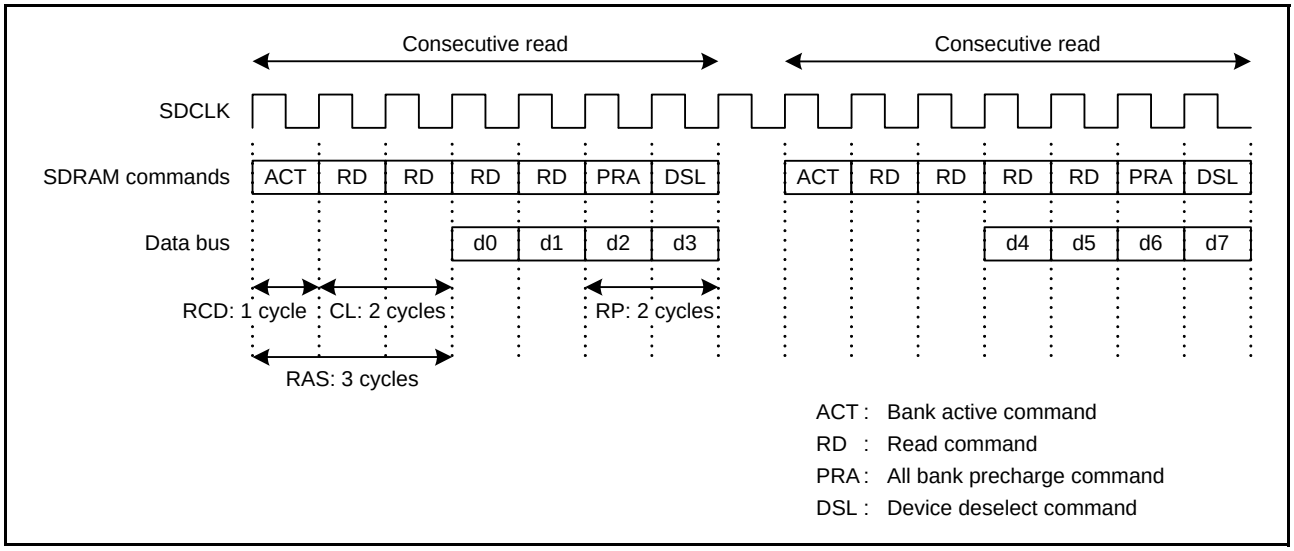


Figure 16.82 Timing Example of Consecutive Read (1)

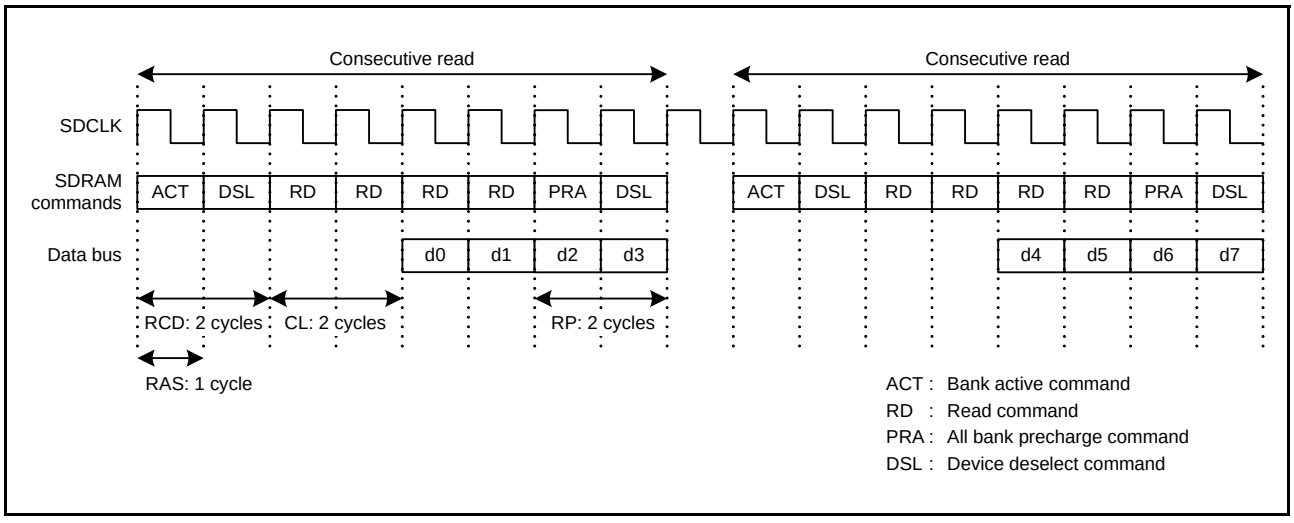


Figure 16.83 Timing Example of Consecutive Read (2)

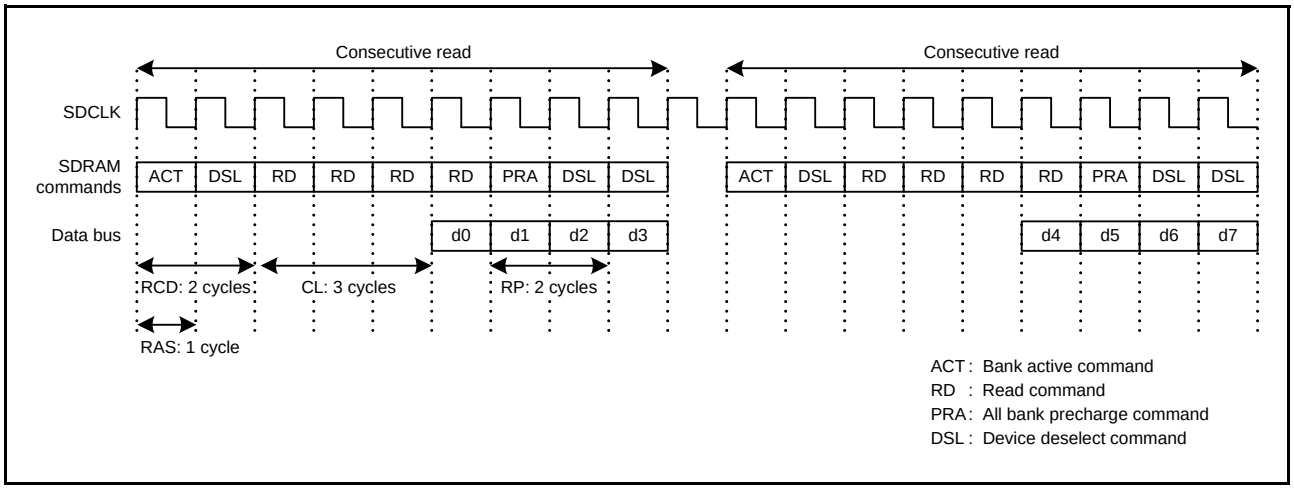


Figure 16.84 Timing Example of Consecutive Read (3)

(4) Consecutive Write Timing Examples

Figure 16.85 to Figure 16.87 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 STDR 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 16.85	010	3	00	1	001	2	0	1
Figure 16.86	000	1	01	2	001	2	0	1
Figure 16.87	000	1	01	2	001	2	1	2

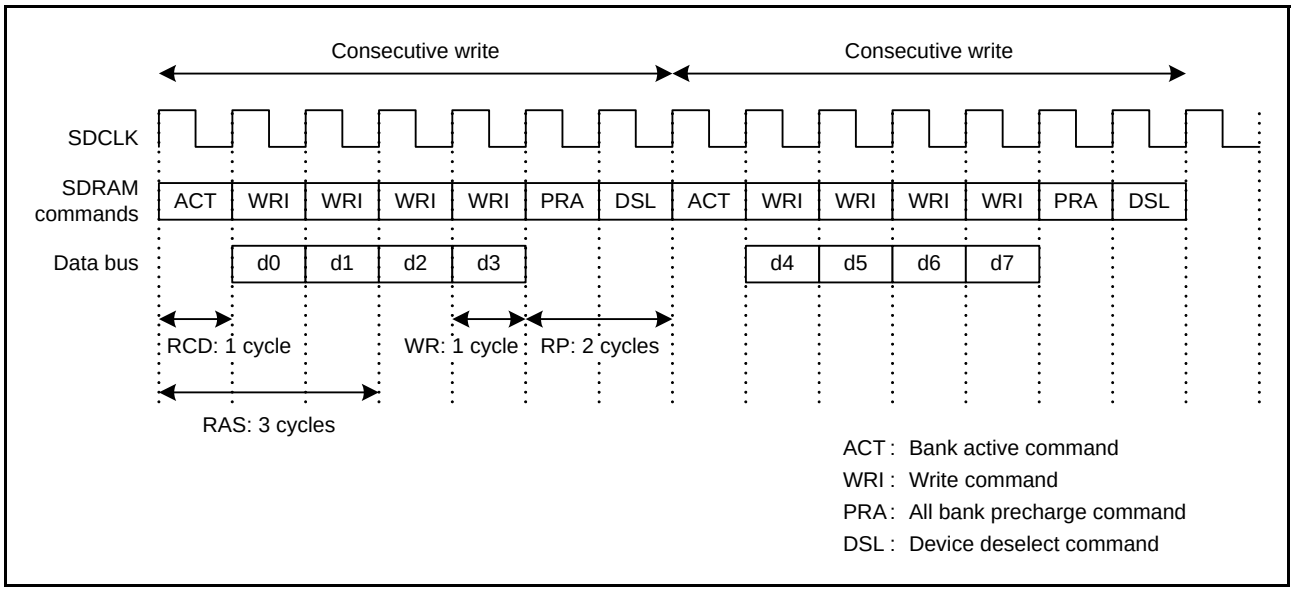


Figure 16.85 Timing Example of Consecutive Write (1)

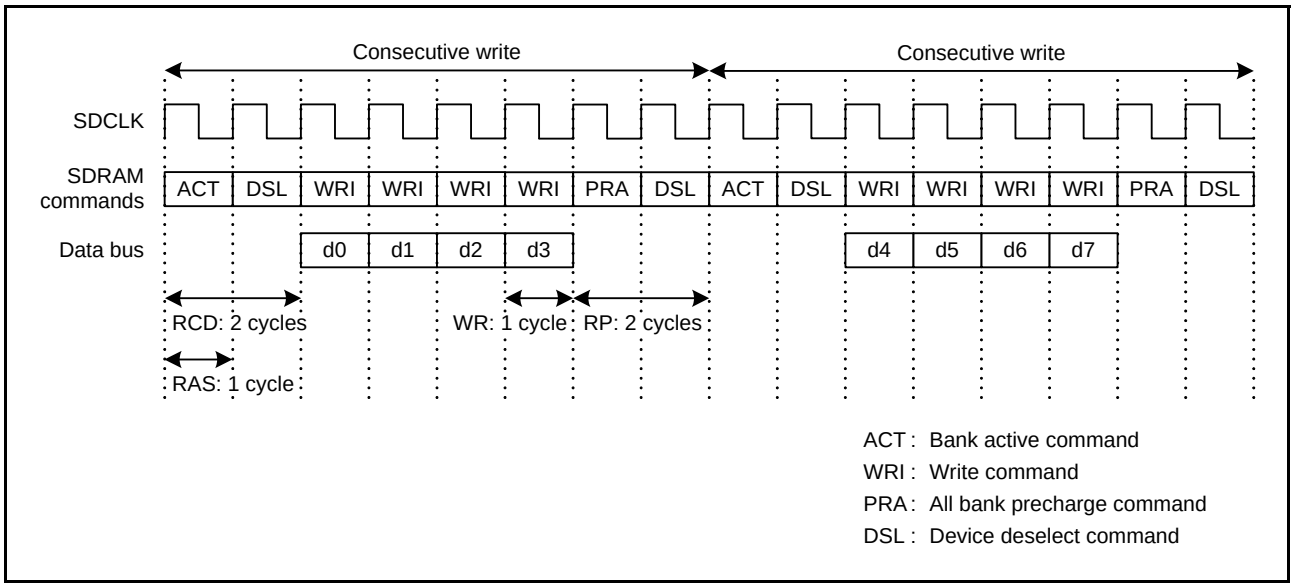


Figure 16.86 Timing Example of Consecutive Write (2)

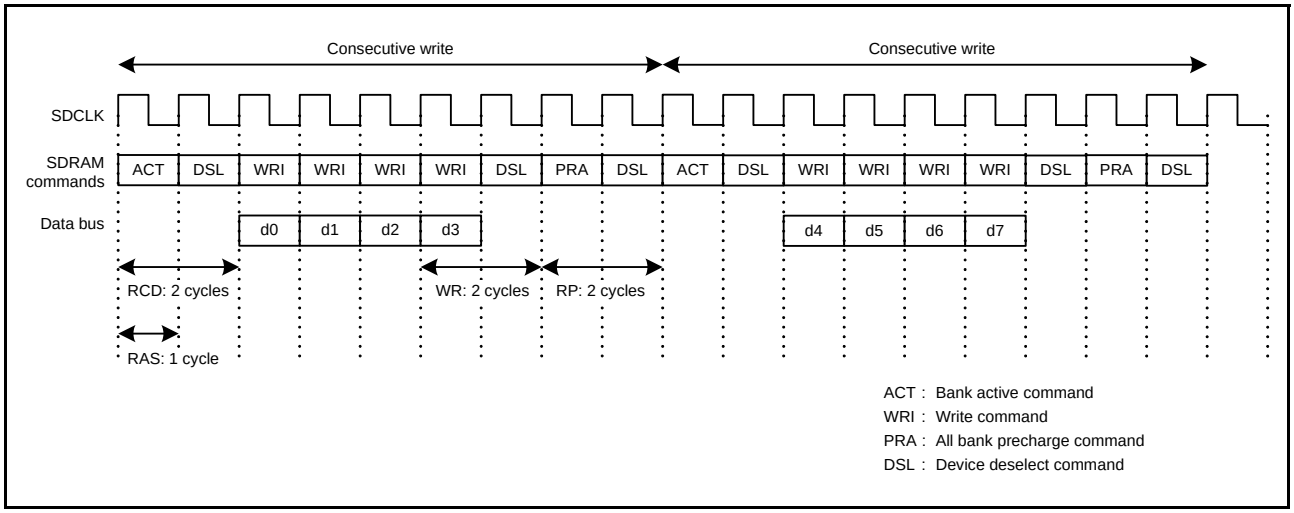


Figure 16.87 Timing Example of Consecutive Write (3)

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

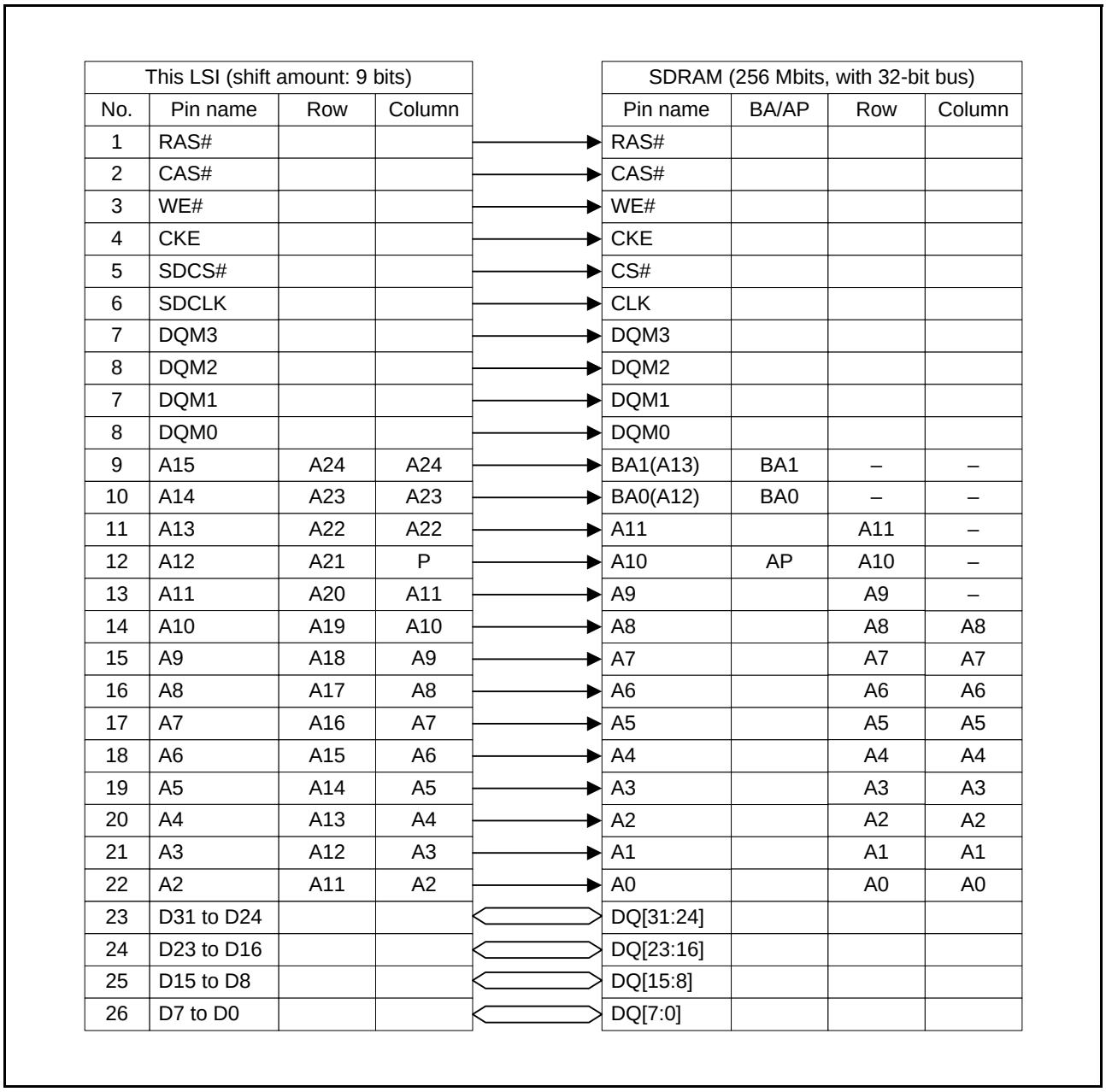
16.6.14.1 32-Bit Bus Space

Figure 16.88 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 16.88 SDRAM Connection Example (512-Mbit x 2, with 16-Bit Bus)

Figure 16.89 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 16.89 SDRAM Connection Example (256-Mbit x 1, with 32-Bit Bus)

Figure 16.90 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 16.90 SDRAM Connection Example (128-Mbits x 2, with 16-Bit Bus)

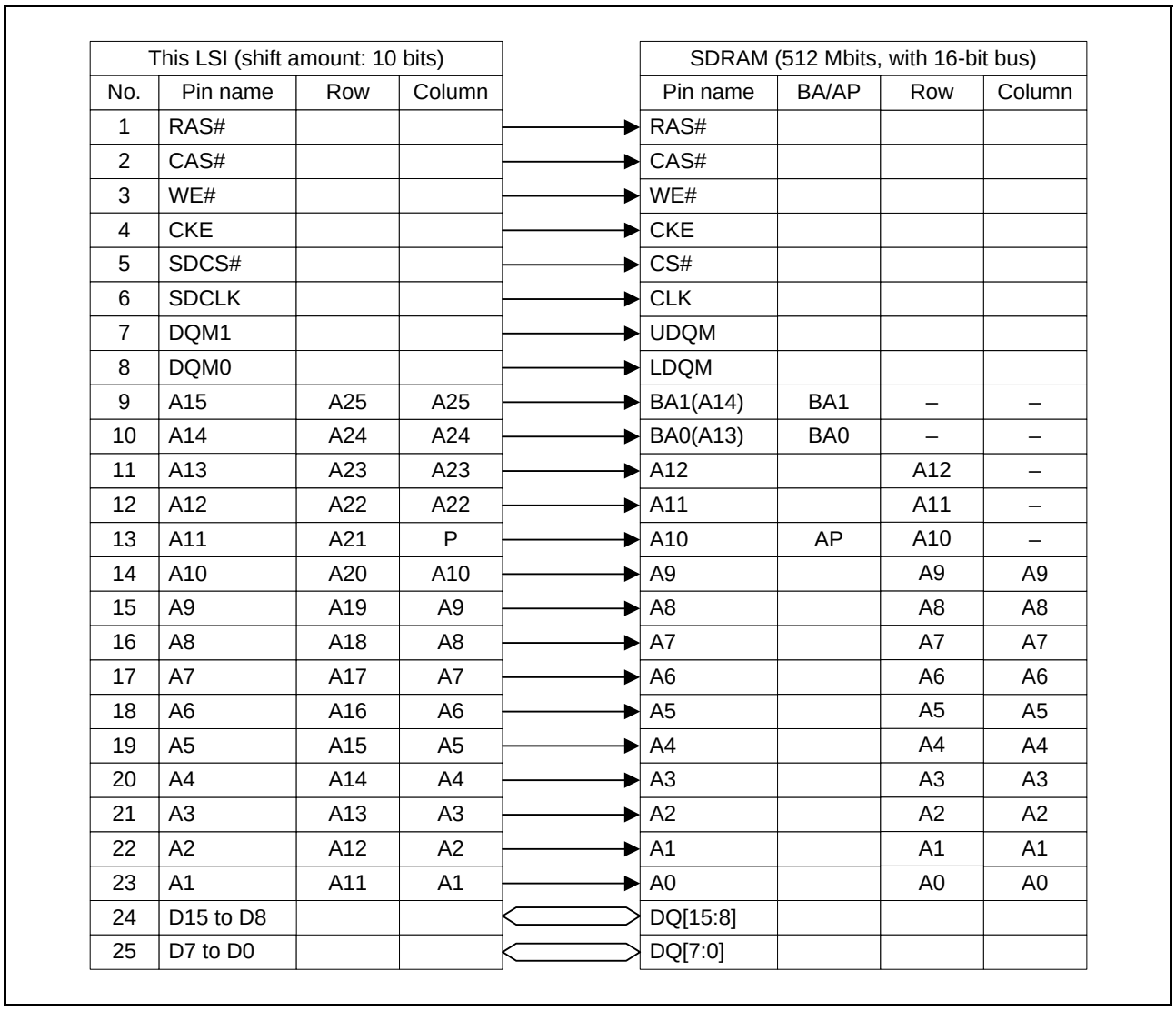
16.6.14.2 16-Bit Bus Space

Figure 16.91 shows an example for connecting to two 512-Mbit SDRAMs with 13-bit row address, 11-bit column address and 8-bit bus.

This LSI (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 16.91 SDRAM Connection Example (512-Mbit x 2, with 8-Bit Bus)

Figure 16.92 shows an example for connecting to a 512-Mbit SDRAM 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	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 16.92 SDRAM Connection Example (512-Mbit x 1, with 16-Bit Bus)

Figure 16.93 shows an example for connecting to a 256-Mbit SDRAM with 13-bit row address, 9-bit column address and 16-bit bus.

This LSI (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 16.93 SDRAM Connection Example (256-Mbit x 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.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKB cycles.
- Internal peripheral buses (4 and 5): 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.
- 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.

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

	Type of Area		Type of Error			
			Illegal Address Access		Timeout	
	On-chip ROM Mode		On-chip ROM Mode		On-chip ROM Mode	
Address	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 00FF FFFFh	Internal peripheral bus 6	Reserved area	Δ	○	—	—
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 on-chip RAM, DataFlash, and on-chip ROM differs depending on the product. For details, see section 44, RAM, section 45, ROM (Flash Memory for Code Storage), and section 46, E2 DataFlash Memory (Flash Memory for Data Storage).

17. DMA Controller (DMACA)

The RX63N/RX631 Group incorporates a 4-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.

17.1 Overview

Table 17.1 lists the specifications of the DMAC, and Figure 17.1 shows a block diagram of the DMAC.

Table 17.1 Specifications of DMAC

Item		Description
Number of channels		4 (DMACm (m = 0 to 3))
Transfer space		512 Mbytes (0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		1 Mbyte (Maximum number of transfers in block transfer mode: 1,024 data × 1,024 blocks)
DMA request source		• Activation 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 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 two bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Power consumption reduction function		Module stop state can be set.

Note 1. For details on DMAC activation sources, see Table 15.3, Interrupt Vector Table in section 15, Interrupt Controller (ICUb).

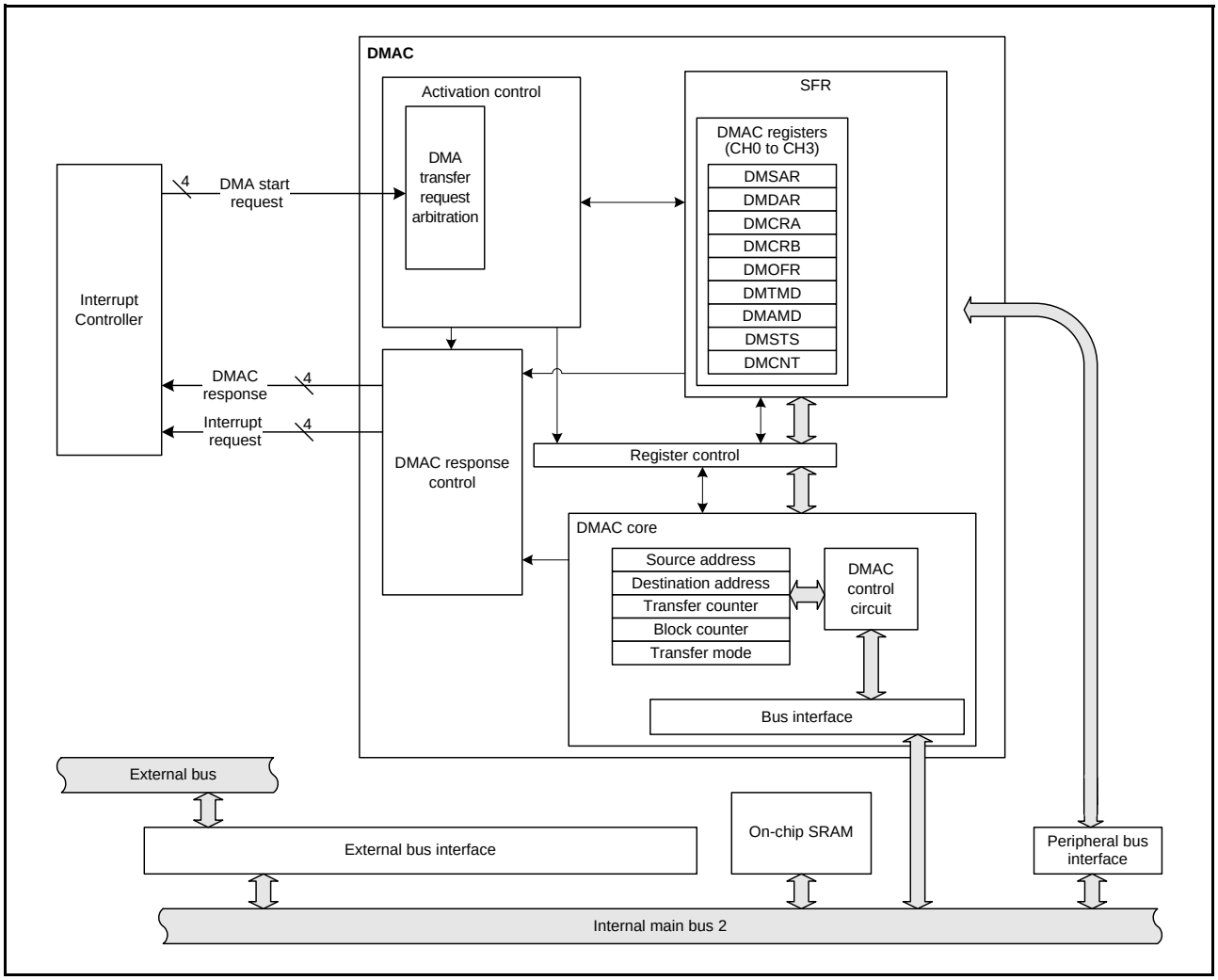


Figure 17.1 Block Diagram of DMAC

17.2 Register Descriptions

17.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h
DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h

	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

Set DMSAR while DMAC activation is disabled (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.

17.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h
DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h

	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

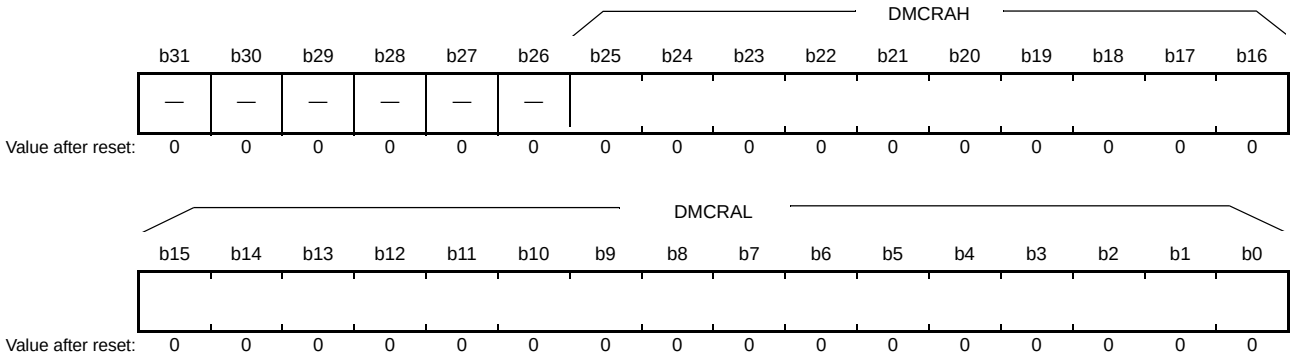
Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set DMDAR while DMAC activation is disabled (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.

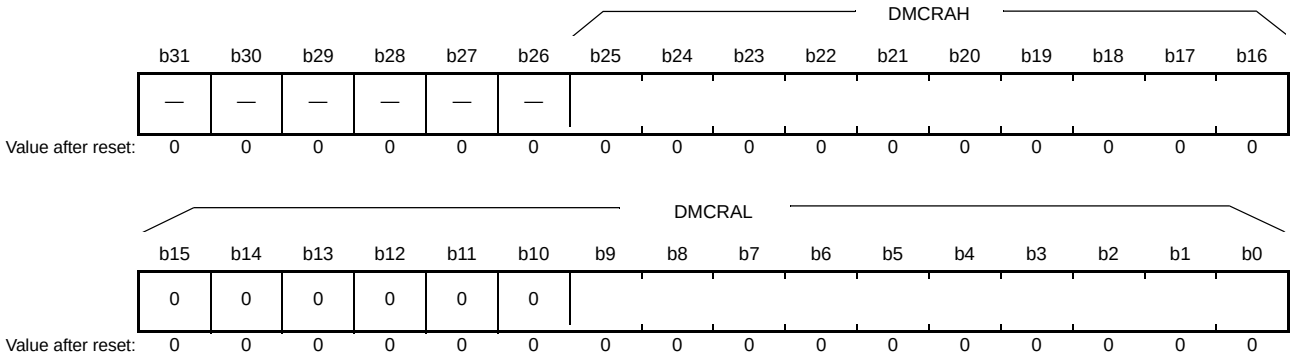
17.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h
DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h

- Normal transfer mode



- Repeat transfer mode, block transfer mode



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 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).
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.

17.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch
DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh

	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 number of block transfer operations or repeat transfer operations.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMCRB specifies the number of block transfer operations and repeat transfer operations in block and repeat transfer mode, respectively.
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, 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.

17.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h
DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h

	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]	DMA 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. DMAC activation source is selected using the DMRSRm registers of the ICU. For details on DMAC activation sources, see Table 15.3, Interrupt Vector Table in section 15, Interrupt Controller (ICUb).

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.

17.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h
DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h

	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 cleared 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 cleared 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 cleared 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 cleared 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 clearing 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 clearing this bit or the DTIF bit in DMSTS to 0.

17.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h
DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h

	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 17.2.	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 17.2.	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: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination 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, or 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 17.2 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 21 (2 bytes) and 217 (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, or 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 17.2 lists the settings and the corresponding extended repeat areas.

SM Bit (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 17.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas

SARA4 to SARA0 or DARA4 to DARA0	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.

17.2.8 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.

17.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch
DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh

	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 activation is enabled) 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.

17.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh
DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh

	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 cleared to 0 if the CLRS bit is set to 0. This bit is not cleared 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 activation 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 cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

17.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh
DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh

	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

17.2.12 DMA Activation Source Flag Control Register (DMCSL)

Address(es): DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh
DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh

	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 flag of the activation source to 0. 1: At the end of transfer, the interrupt flag of the activation 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 flag of the activation source of the DMAC is cleared to 0 or issues an interrupt to the CPU, at the beginning of transfer.
When DMTMD.DCTG[1:0] = 00b (activation by software), the setting of the DISEL bit does not affect the operation.

17.2.13 DMACA Module Activation 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 Operation Enable	0: DMAC activation is disabled. 1: DMAC activation is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMST Bit (DMAC Operation Enable)

When this bit is set to 1, DMAC activation is enabled 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 activation is enabled), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is cleared 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 DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

17.3 Operation

17.3.1 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 17.3 summarizes the register update operation in normal transfer mode, and Figure 17.2 shows the operation in normal transfer mode.

Table 17.3 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 count	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.

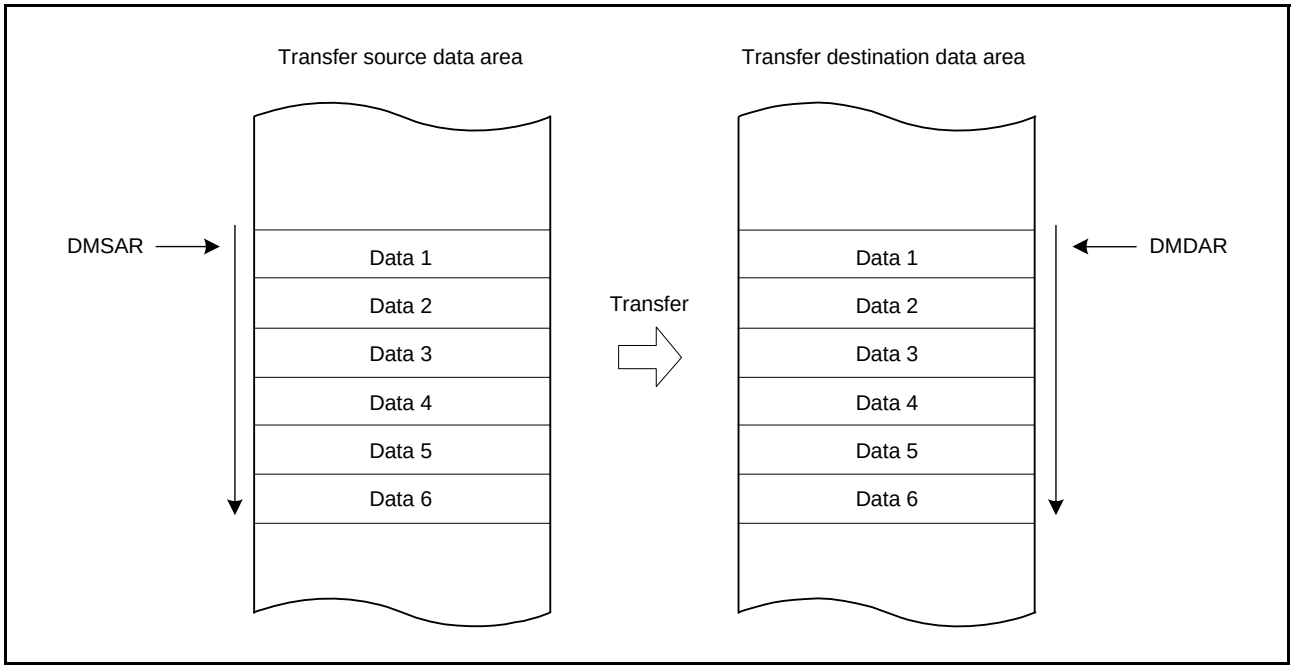


Figure 17.2 Operation in Normal Transfer Mode

(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 1K can be set as the number of repeat transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count 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 17.4 summarizes the register update operation in repeat transfer mode, and Figure 17.3 shows the operation in repeat transfer mode.

Table 17.4 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 count	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Count of repeat transfer operations	Not updated	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

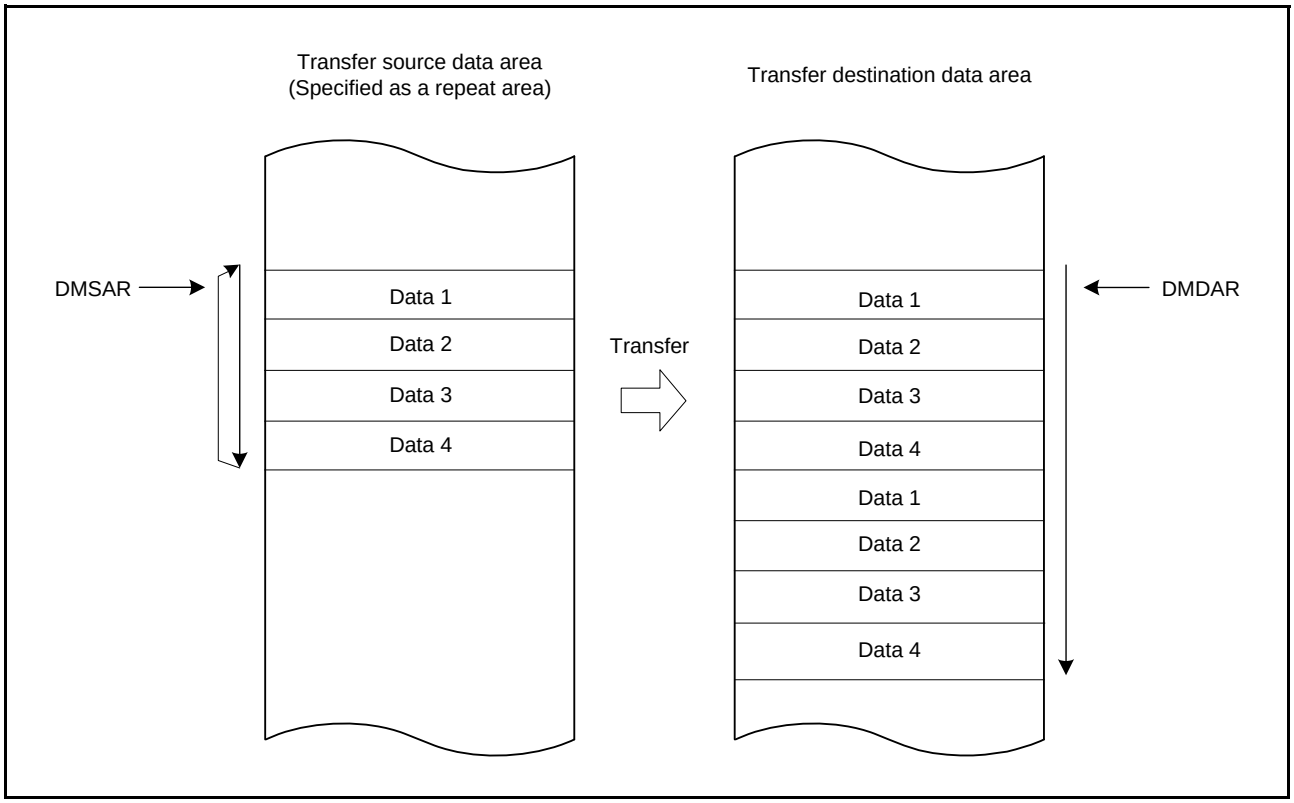


Figure 17.3 Operation in Repeat Transfer Mode

(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 1K can be set as the number of block transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count 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 17.5 summarizes the register update operation in block transfer mode, and Figure 17.4 shows the operation in block transfer mode.

Table 17.5 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	Transfer count	DMACm.DMCRAH
DMACm.DMCRB	Count of block transfer operations	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

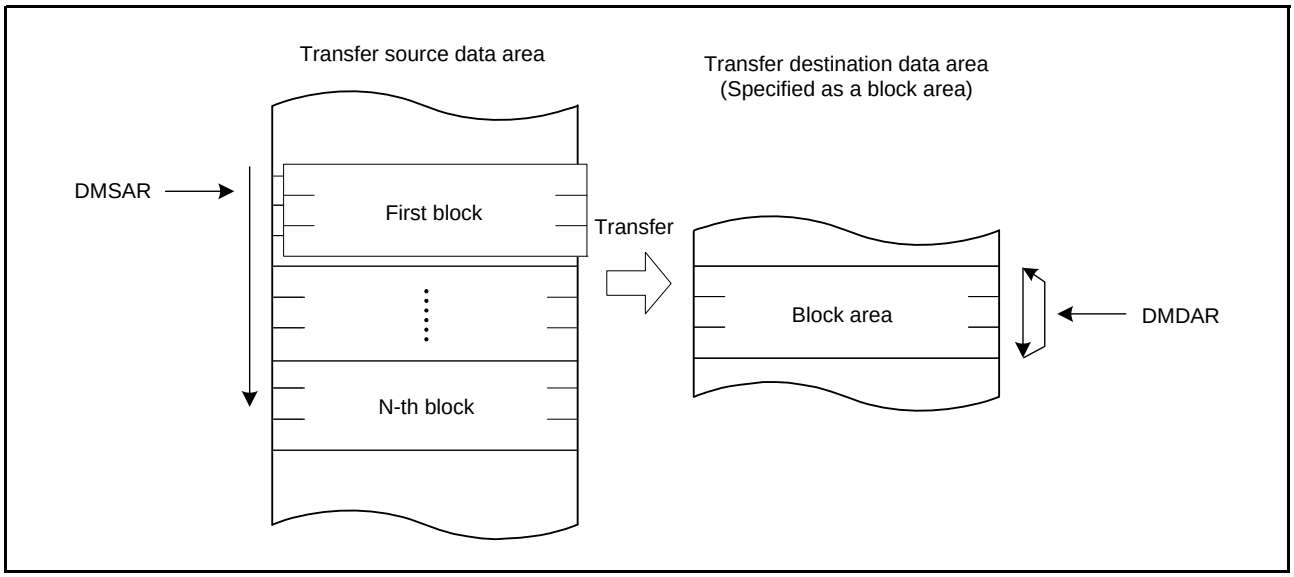


Figure 17.4 Operation in Block Transfer Mode

17.3.2 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 cleared 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 17.5 shows an example of the extended repeat area operation.

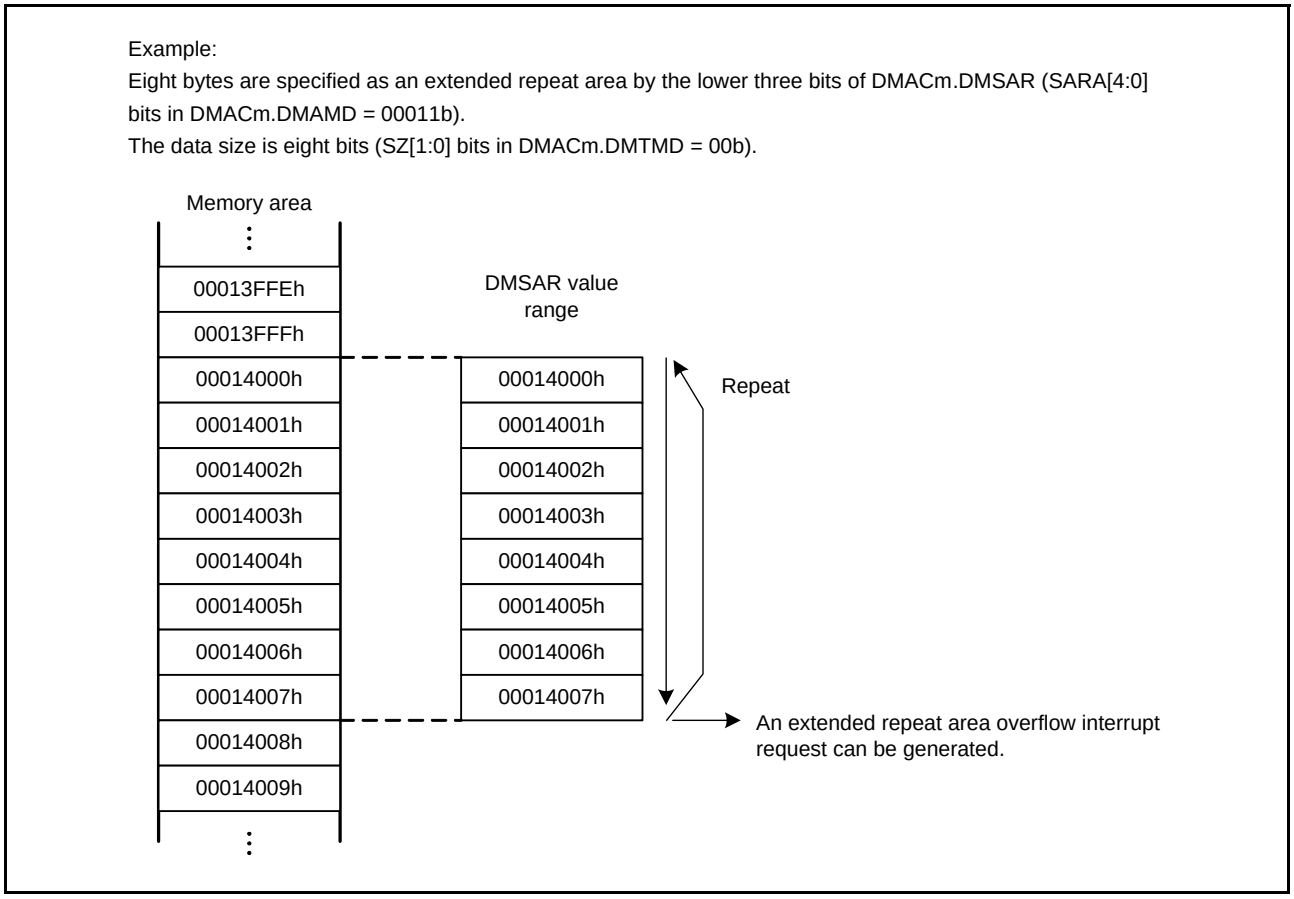


Figure 17.5 Example of Extended Repeat Area Operation

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 17.6 shows an example when the extended repeat area function is used in block transfer mode.

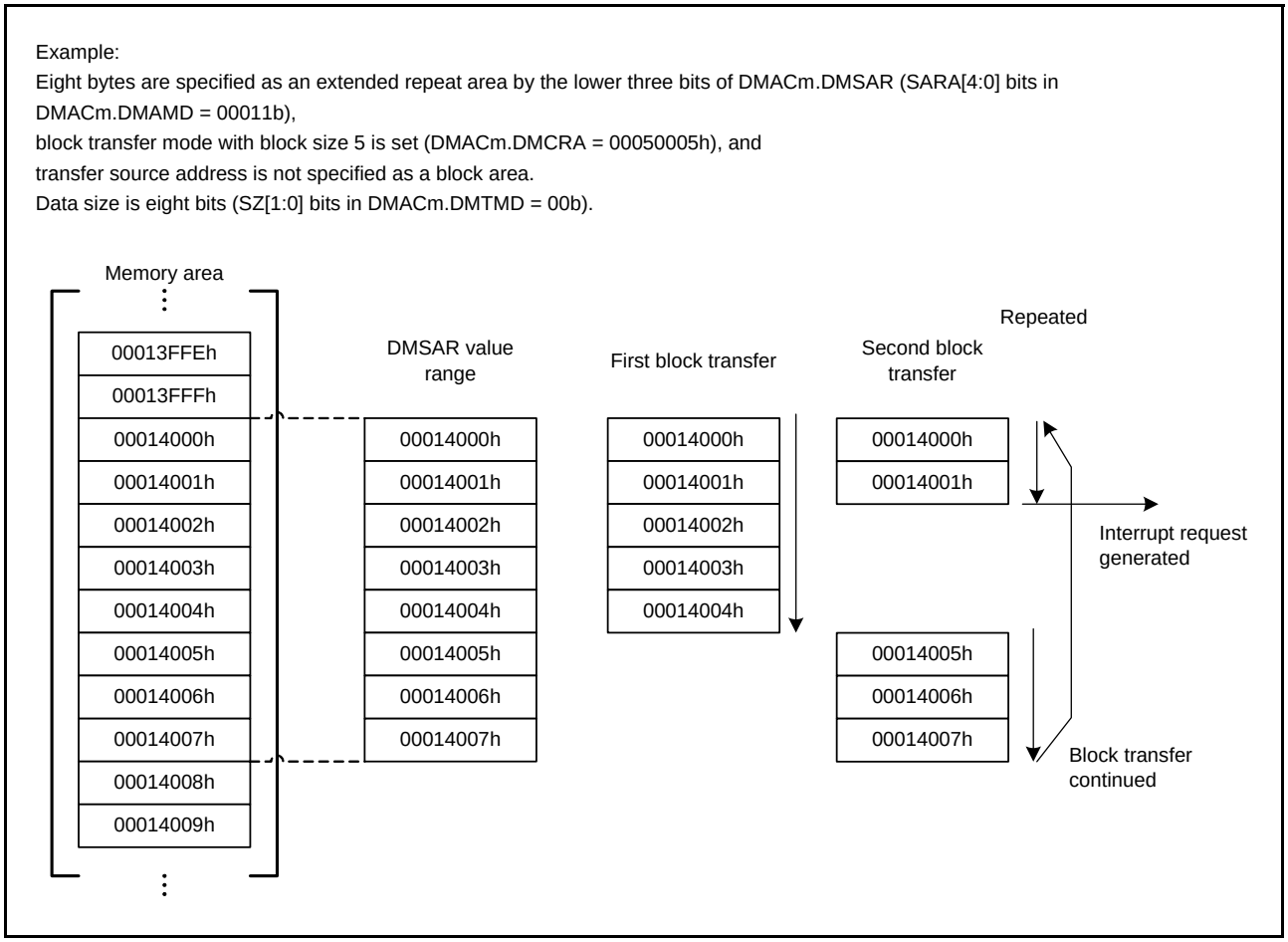


Figure 17.6 Example of Extended Repeat Area Function in Block Transfer Mode

17.3.3 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 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 17.6 shows the address update method in each address update mode.

Table 17.6 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 = 1 + ~offset (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 17.7 shows an example of address updating using offset addition.

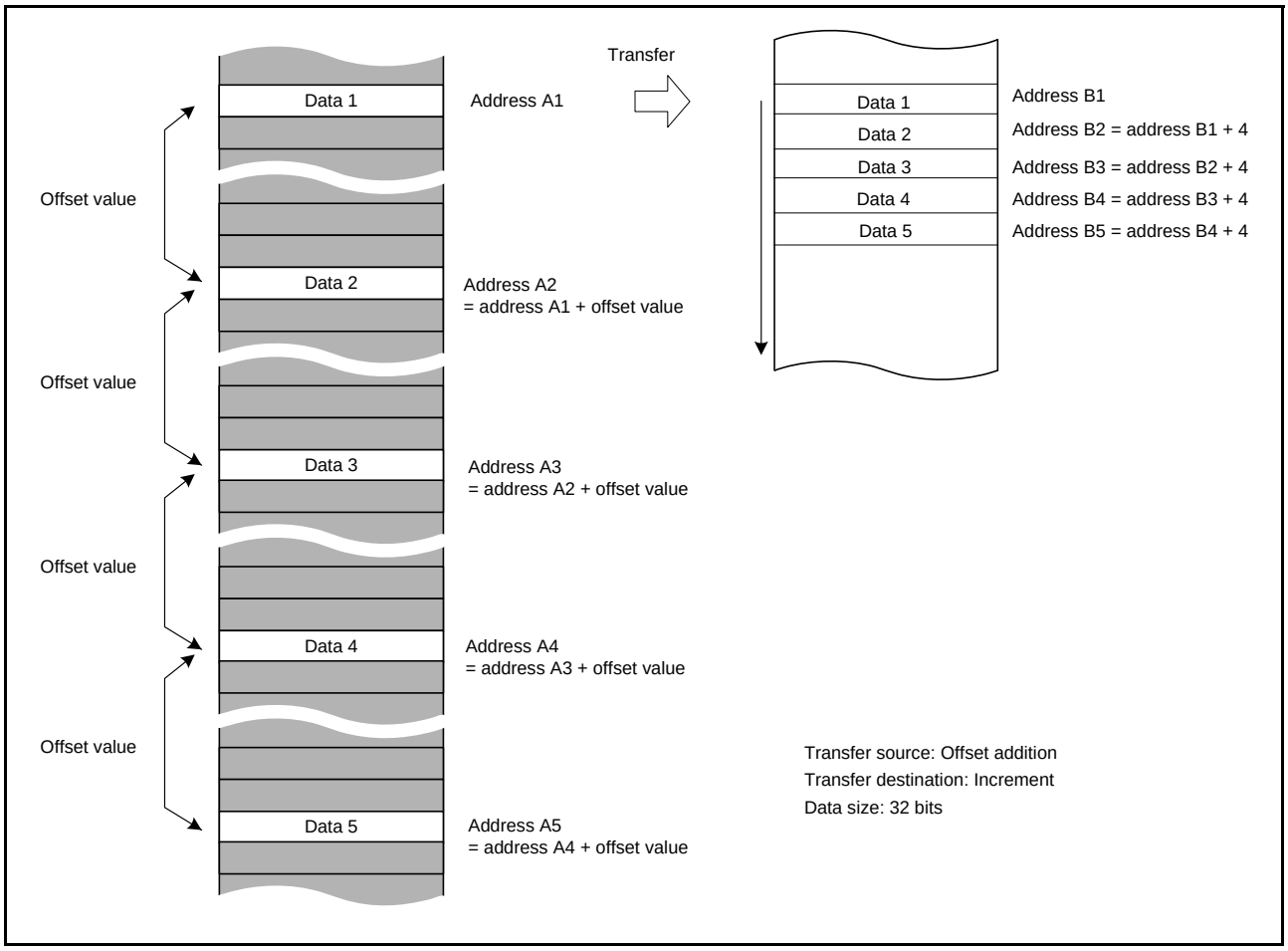


Figure 17.7 Example of Address Updating by Offset Addition

In Figure 17.7, 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 17.8 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.

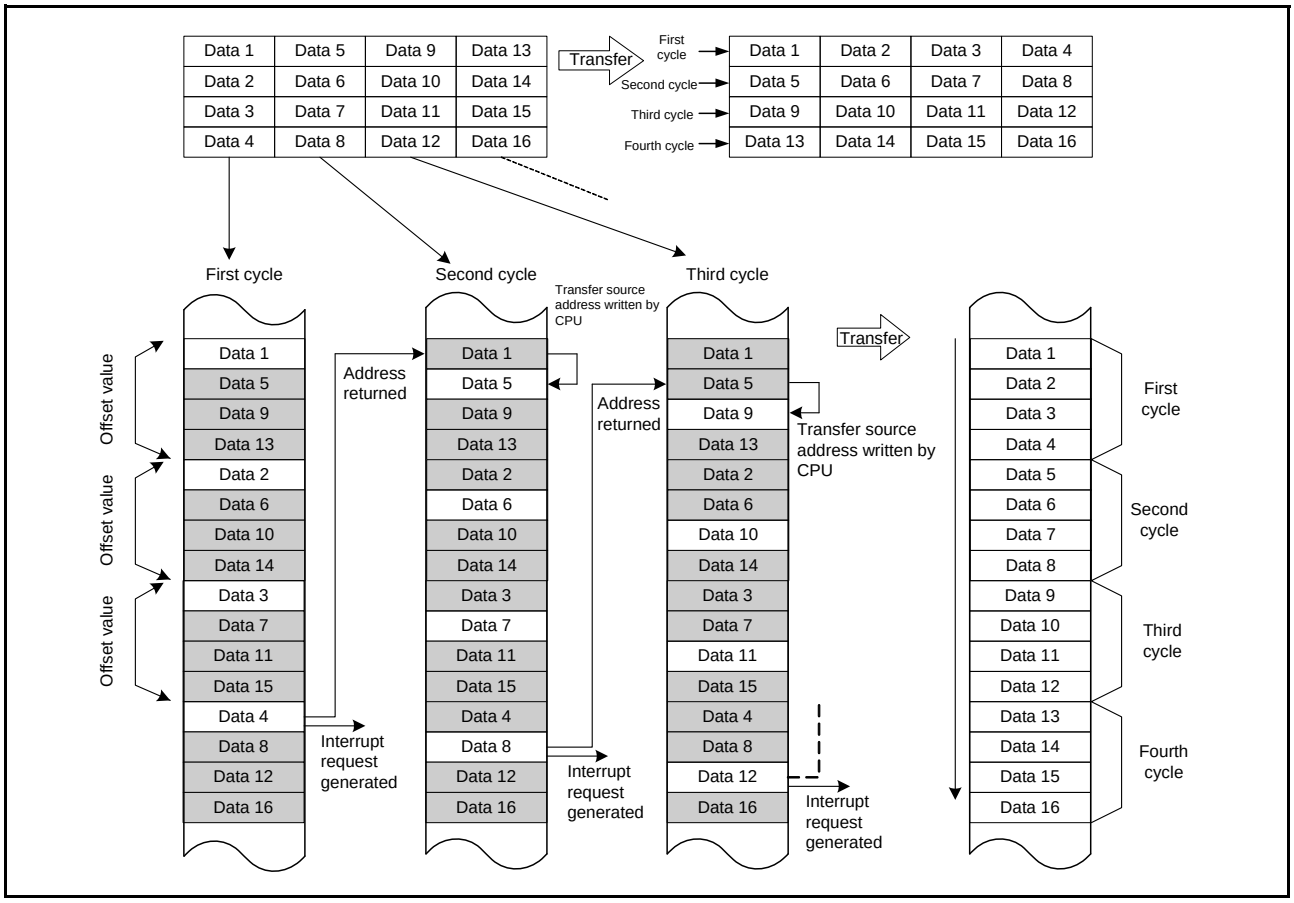


Figure 17.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

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 followings.

- 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 17.9 shows a flowchart of the XY conversion.

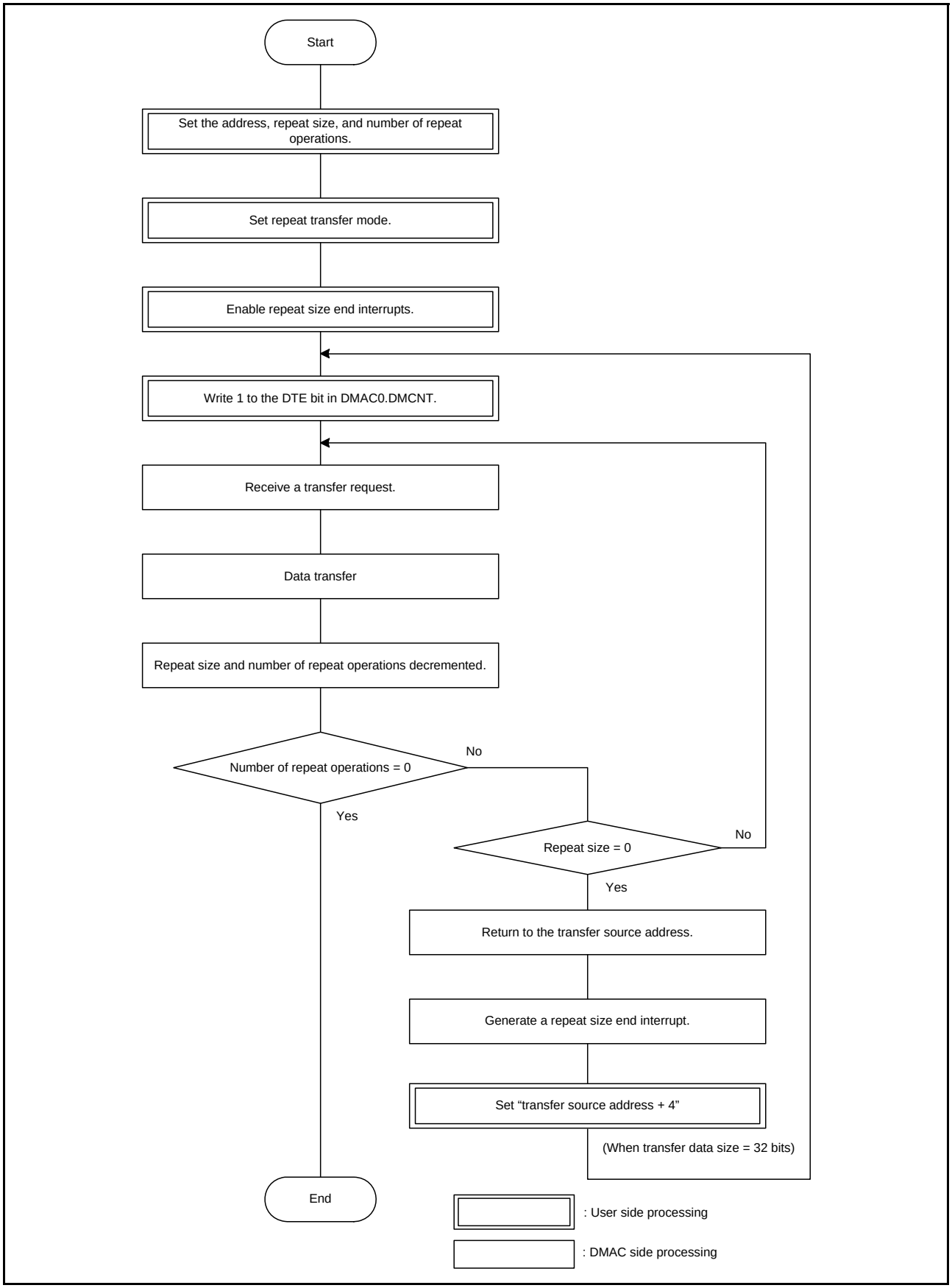


Figure 17.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

17.3.4 Starting DMA Transfer

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMAC activation sources. Setting the DCTG[1:0] bits in DMTMD of DMACm selects the activation source.

(1) DMAC Activation by Software

Setting the DCTG[1:0] bits in DMTMD of DMACm to 00b enables the DMAC activation 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 activation enabled).
When the DMAC is activated by software while the CLRS bit in DMREQ of DMACm is 0, the SWREQ bit in DMREQ of DMACm is cleared to 0 after data transfer is started in response to a DMA transfer request.
When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) DMAC Activation 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 DMAC activation sources. The activation source can be selected separately for each channel using the DMRSRm registers (m = 0 to 3) of the ICU.
The DMAC is activated 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 is 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 activation is enabled).

For interrupt requests specified as DMAC activation sources, see Table 15.3, Interrupt Vector Table, in section 15, Interrupt Controller (ICUb).

17.3.5 Operation Timing

Figure 17.10 and Figure 17.11 show DMAC operation timing examples.

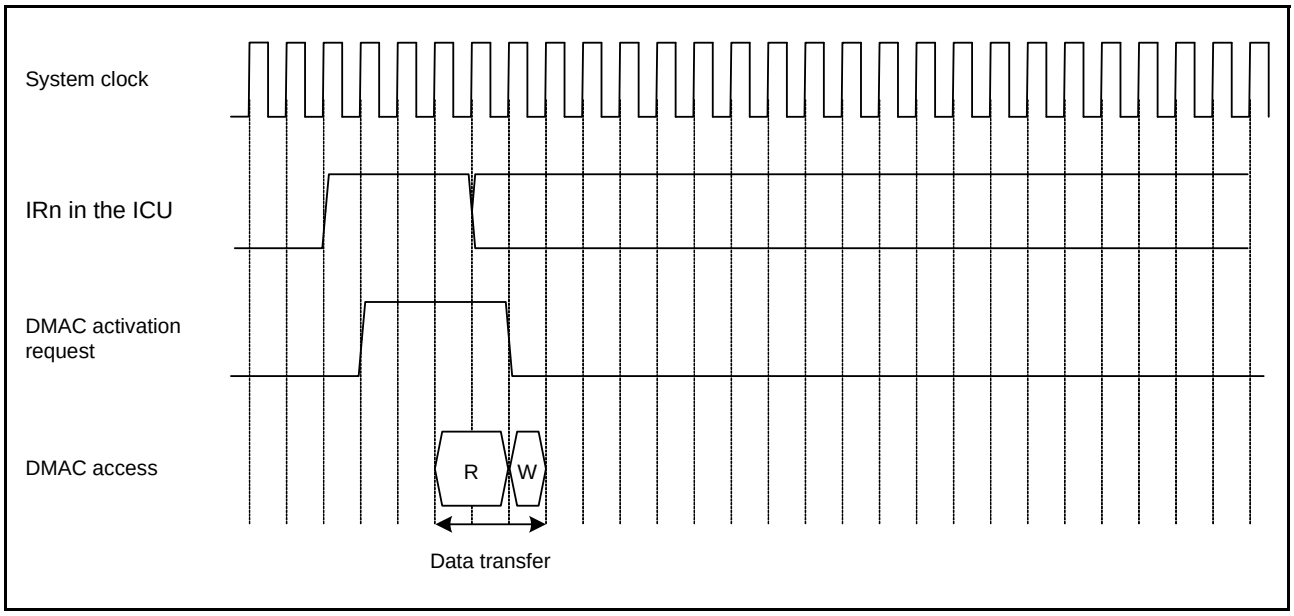


Figure 17.10 DMAC Operation Timing Example (1) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

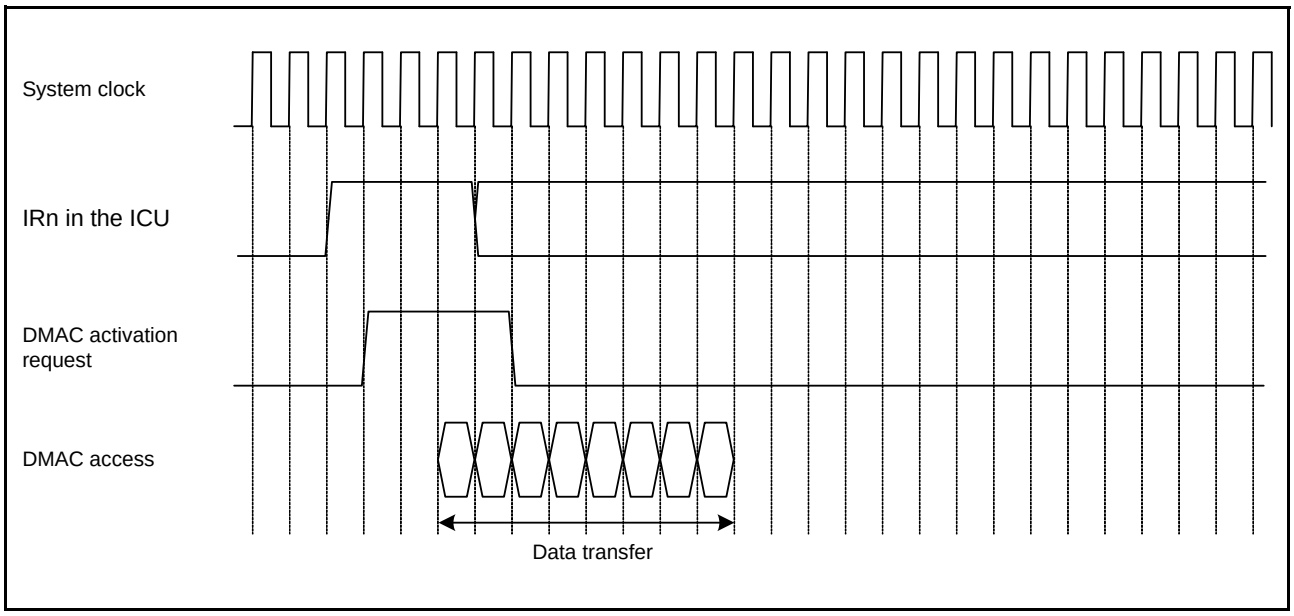


Figure 17.11 DMAC Operation Timing Example (2) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

17.3.6 DMAC Execution Cycles

Table 17.7 lists execution cycles in one DMAC data transfer operation.

Table 17.7 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 44, RAM, section 45, ROM (Flash Memory for Code Storage), section 5, I/O Registers, and section 16.2.5, External Bus.
The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).
For the operation example, see section 17.3.5, Operation Timing.

17.3.7 Activating the DMAC

Figure 17.12 shows the register setting procedure.

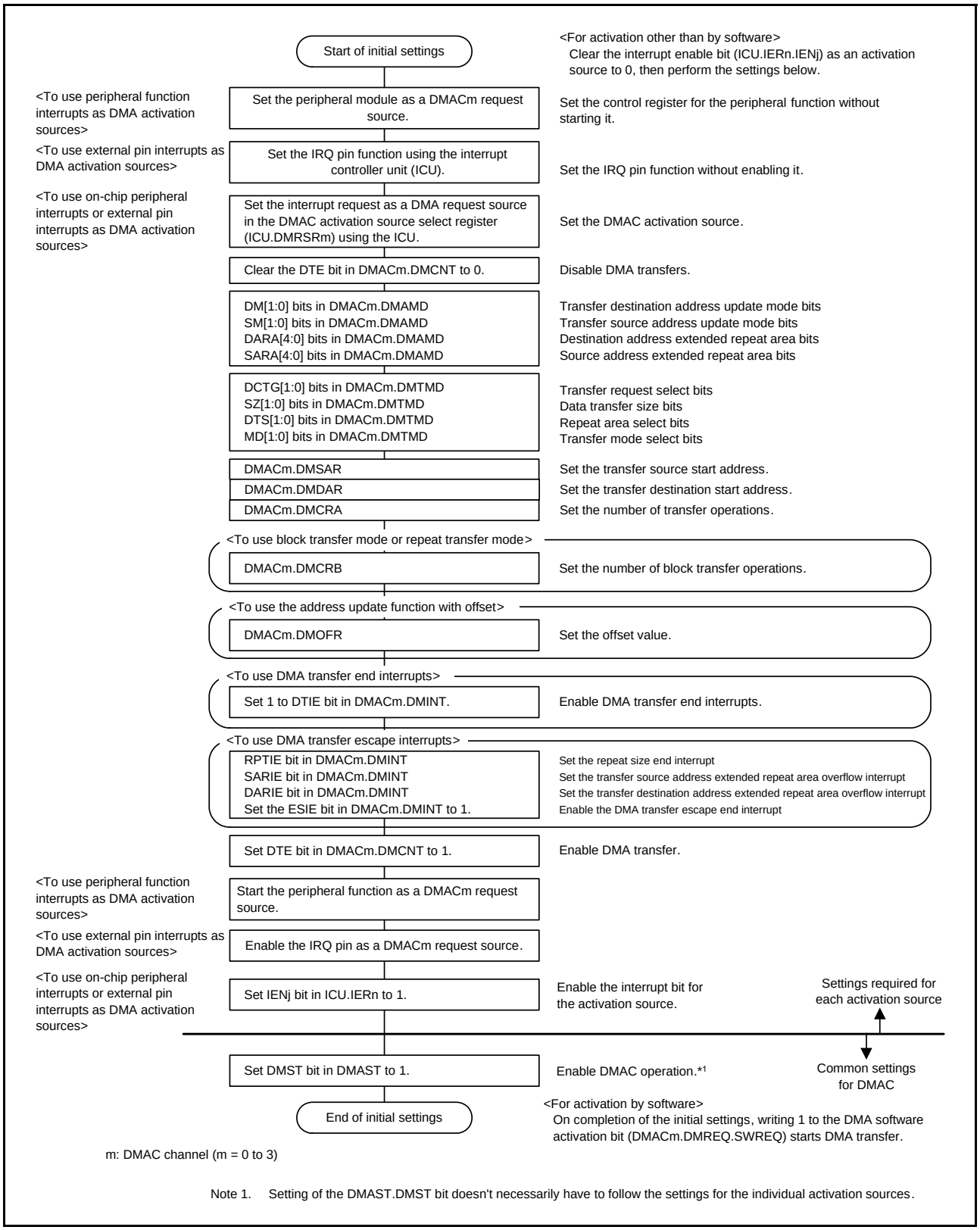


Figure 17.12 Register Setting Procedure

17.3.8 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 start enabled) enable DMA transfer of channel m (m = 0 to 3).
Another activation 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).

17.3.9 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 17.3 to Table 17.5.

(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 17.3 to Table 17.5.

(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 17.3 to Table 17.5.

(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 17.3 to Table 17.5.

(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 cleared 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 cleared 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 cleared 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 cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared 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 cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared 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 (ICUb).

17.3.10 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 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.

17.4 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.

17.4.1 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 cleared 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 cleared 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 cleared 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 (ICUb).

17.4.2 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 cleared 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 (ICUb).

17.4.3 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 cleared 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 (ICUb).

17.5 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 17.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 17.13 shows the schematic logic diagram of interrupt outputs. Figure 17.14 shows the DMAC interrupt handling routine to resume or terminate DMA transfer.

Table 17.8 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		

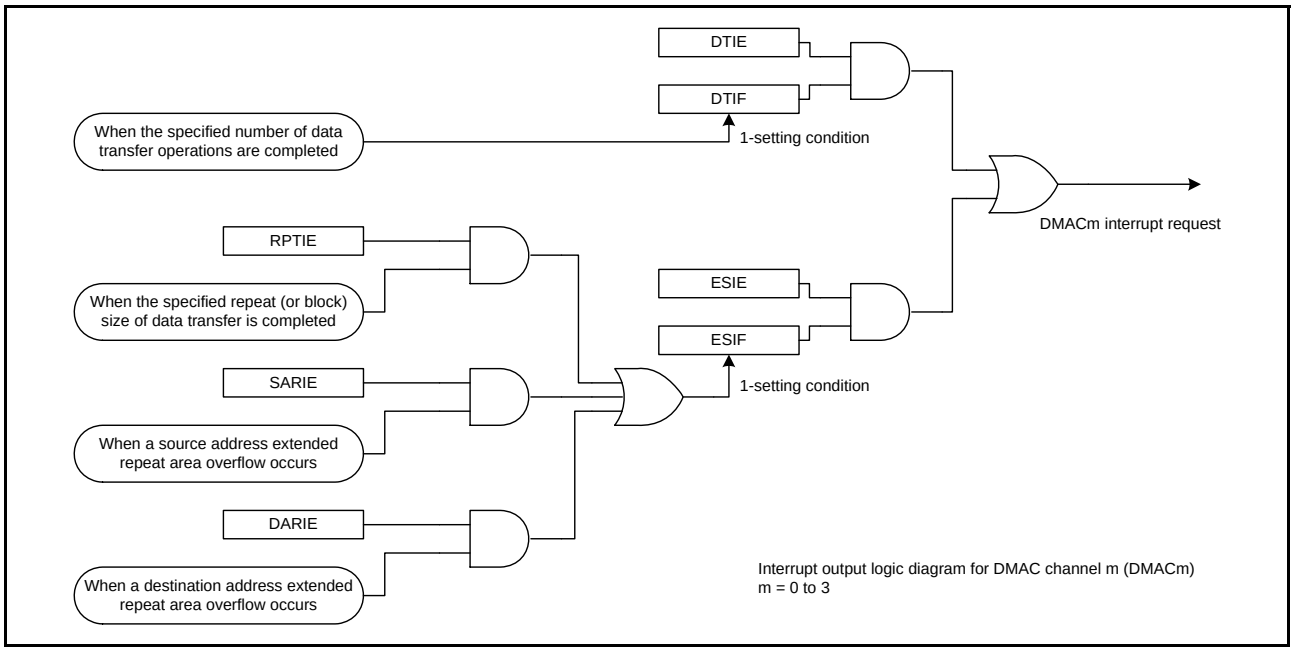


Figure 17.13 Schematic Logic Diagram of Interrupt Outputs

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 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 cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

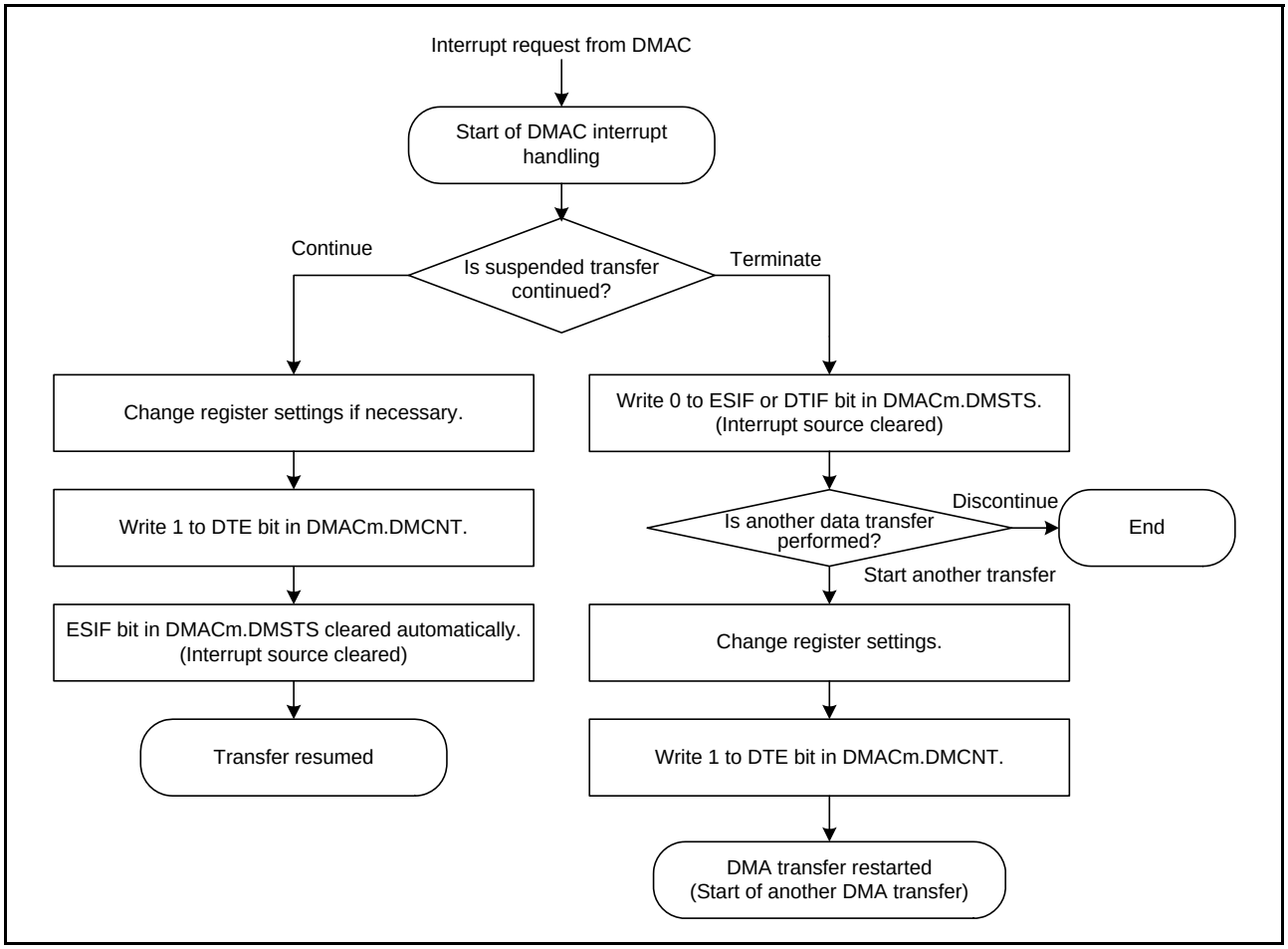


Figure 17.14 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

17.6 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 15.5.3, Selecting Interrupt Request Destinations in section 15, Interrupt Controller (ICUb), and then execute the WAIT instruction.

17.7 Usage Notes

17.7.1 DMA Transfer to External Devices

In DMA transfer to an external device, the ACT bit in DMSTS of DMACm may be cleared to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the external bus access.

17.7.2 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the ACT bit in DMSTS of DMACm may be cleared to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

17.7.3 Access to the Registers during DMA Transfer

The DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACm must not be accessed 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).

17.7.4 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.

17.7.5 Interrupt Request by the DMA Activation Source Flag Control Register (DMCSL) at the End of each Transfer

While the DMACm.DMCSL.DISEL bit is 1, an interrupt is issued to the CPU at the end of each transfer that has been activated 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 flag of the DMAC activation source to 0 by changing the interrupt request destination to the CPU. In this case, since the interrupt flag is not cleared to 0 at the end of DMAC transfer, it should be cleared 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 (ICUb). For the DMACm.DMCSL.DISEL bit setting, see section 17.2.12, DMA Activation Source Flag Control Register (DMCSL).

17.7.6 Setting of DMAC Activation Source Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC activation source select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMACm.DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation 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 (ICUb).

17.7.7 Suspending or Restarting DMA Activation

To suspend a DMA activation request, write 0 to the interrupt enable bit for the activation source (ICU.IERn.IENj bit). To restart the DMA transfer, write 1 to the ICU.IERn.IENj bit with the setting shown in section 17.3.7, Activating the DMAC.

18. EXDMA Controller (EXDMACa)

The RX63N/RX631 Group 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.

18.1 Overview

Table 18.1 lists the specifications of the EXDMAC, and Figure 18.1 shows a block diagram of the EXDMAC.

Table 18.1 Specifications of EXDMAC

Item		Description
Number of channels		2 (EXDMAC0 and EXDMAC1)
Transfer space		512 Mbytes (00000000h to 0FFFFFFFh and F0000000h to FFFFFFFFh excluding reserved areas)
Maximum transfer volume		1 M data (Maximum number of transfer operations in block transfer mode: 1024 data × 1024 blocks)
DMA request source		• Activation source selectable from the following three sources for each channel Software trigger External DMA transfer request input DMA transfer request from peripheral modules (compare match A of MTU1 or TPU7)
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 1024data
	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: 1024 data
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1024 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 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 two bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	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.

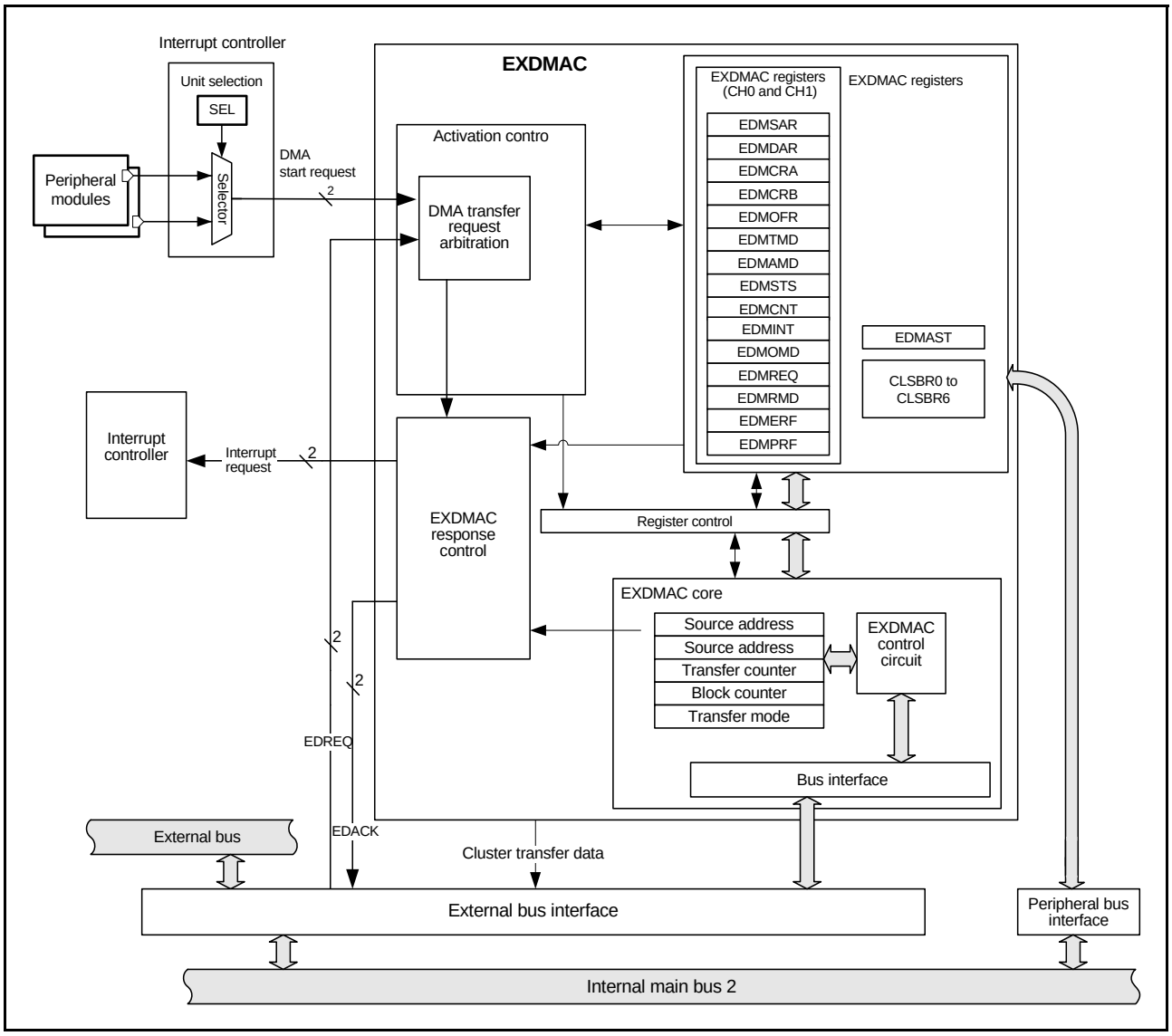


Figure 18.1 Block Diagram of EXDMAC

Table 18.2 lists the input/output pins of the EXDMAC.

Table 18.2 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

18.2 Register Descriptions

18.2.1 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.	00000000h to 0FFFFFFFh (256 Mbytes) F0000000h to FFFFFFFFh (256 Mbytes)	R/W

Set EDMSAR while EXDMAC activation is disabled (the DMST bit in EDMAST = 0) or DMA transfer is disabled (the DTE bit in EDMCNT = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading EDMSAR returns the extended value.
EDMSAR must be accessed in 32 bits.

18.2.2 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.	00000000h to 0FFFFFFFh (256 Mbytes) F0000000h to FFFFFFFFh (256 Mbytes)	R/W

Set EDMDAR while EXDMAC activation is disabled (the DMST bit in EDMAST = 0) or DMA transfer is disabled (the DTE bit in EDMCNT = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading EDMDAR returns the extended value.
EDMDAR must be accessed in 32 bits.

18.2.3 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.

(1) Normal transfer mode (MD[1:0] bits in EXDMACn.EDMTMD = 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 (MD[1:0] bits in EXDMACn.EDMTMD = 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 (MD[1:0] bits in EXDMACn.EDMTMD = 10b)

EDMCRAH specifies the block size and EDMCRAL functions as a 10-bit block size 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 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 (MD[1:0] bits in EXDMACn.EDMTMD = 11b)

EDMCRAH specifies the cluster size and EDMCRAL functions as a 3-bit cluster size counter.
The number of transfer operations is one when the setting is 001h, seven when it is 007h, and eight 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.

18.2.4 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 number of block transfer operations in block transfer mode, the number of repeat transfer operations in repeat transfer mode, or the number of cluster transfer operations 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

The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h.
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, a value of 3FFh should be set.

18.2.5 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]	DMA Request Source Select	b1 b0 0 0: Software 0 1: Setting prohibited 1 0: External DMA transfer request pin (EDREQn) 1 1: DMA transfer requests from the peripheral modules (compare match A of MTU1 or TPU7)	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

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.

18.2.6 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	EDACK 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

DACKSEL (EDACKn Pin Toggling Selection) Bit

This bit is used to set prohibition or enabling of toggled output 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 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.

During the above transfer in single address mode (AMS bit in EDMAMD = 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).

18.2.7 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

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 EDMCNT is cleared to 0 (DMA transfer is disabled). At the same time, the ESIF bit in EDMSTS 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 DTE bit in EDMCNT 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 cleared to 0 (DMA transfer is disabled). At the same time, the ESIF bit in EDMSTS 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 DTE bit in EDMCNT 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 EDMCNT is cleared to 0 (DMA transfer is disabled) after completion of a 1-repeat size data transfer. At the same time, the ESIF bit in EDMSTS 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 EDMTMD 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 EDMCNT is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF bit in EDMSTS 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 EDMTMD are 10b (= repeat area or block area is not specified.)
When this bit is set to 1 in cluster transfer mode, the DTE bit in EDMCNT is cleared to 0 after completion of a 1-cluster data transfer in the same way as repeat transfer mode. At the same time, the ESIF bit in EDMSTS 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 EDMTMD 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 bit in EXDMACn.EDMSTS is set to 1 with this bit set to1.
The transfer escape end interrupt is cleared by clearing this bit or the ESIF bit in EDMSTS 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 EDMSTS is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DTIF bit in EDMSTS to 0.

18.2.8 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	Specifies the extended repeat area on the destination address. For details on the settings, see tTable 18.3.	R/W
b5	—	Reserved	These bits are always 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 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 18.3.	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	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.

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 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 or underflow occurs in the extended repeat area with the DARIE bit in EDMINT set to 1. Table 18.3 shows 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 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 EXDMAC 0.EDMOFR register is added to the address. Offset addition can be specified only for EXDMAC0.

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 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 or underflow occurs in the extended repeat area with the SARIE bit in EDMINT set to 1. Table 18.3 shows the settings and the corresponding extended repeat areas.

SM[1:0] Bit (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 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 18.3 Settings and Range of Extended Repeat Areas

SARA4 to SARA0 or DARA4 to DARA0	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)

18.2.9 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.	00000000h to 00FFFFFFh (0 bytes to (16M – 1) bytes) FF000000h to FFFFFFFFh (–16M bytes to –1 byte)	R/W

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.

18.2.10 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

DTE Bit (DMA Transfer Enable)

When the DMST bit in EDMAST is set to 1 (EXDMAC activation is enabled) 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 cleared 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.

18.2.11 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

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 cleared to 0 if the CLRS bit is set to 0. This bit is not cleared 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 activation 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 cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

18.2.12 EXDMA Status Register (EDMSTS)

Address(es): EXDMAC0.EDMSTS: 008 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

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 EDMINT set to 1.
- When 1-block data transfer is completed in block transfer mode with the RPTIE bit in EDMINT set to 1.
- When 1-cluster data transfer is completed in cluster transfer mode with the RPTIE bit in EDMINT set to 1.
- When an extended repeat area overflow on the source address occurs while the SARIE bit in EDMINT is set to 1 and the SARA[4:0] bits in EDMAMD 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 EDMINT is set to 1 and the DARA[4:0] bits in EDMAMD 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 EDMCNT.

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 Bit (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

18.2.13 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

DREQS[1:0] Bits (Re guest Input Sense Mode Set)

These bits specify the sense mode for the external DMA transfer request signal (EDREQn pin).

18.2.14 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.

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)

18.2.15 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.

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

18.2.16 EXDMA 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 Operation Enable	0: EXDMAC activation is disabled. 1: EXDMAC activation is enabled.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DMST Bit (EXDMAC Operation Enable)

When this bit is set to 1, EXDMAC activation is enabled 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 cleared 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

18.2.17 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 b1	Buffer area for cluster transfer.	R/W

CLSBRi are buffer registers for cluster transfer. During cluster transfer, transferred data is sequentially stored in CLSBRi 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 CLSBRi is used for all the channels. If a conflict occurs between the write to CLSBRi 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 (SZ[1:0] bits in EDMTMD).

(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 7, 7-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 7, 14-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 7, 28-byte data is one cluster.
Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 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 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 18.4 summarizes the register update operation in normal transfer mode, and Figure 18.2 shows the operation in normal transfer mode.

Table 18.4 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 count	Decremented by one/not updated (in free running mode)
EXDMACn.EDMCRAH	Block size	Not updated (Not used in normal transfer mode)
EXDMACn.EDMCRB	Block count	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for EXDMAC0.

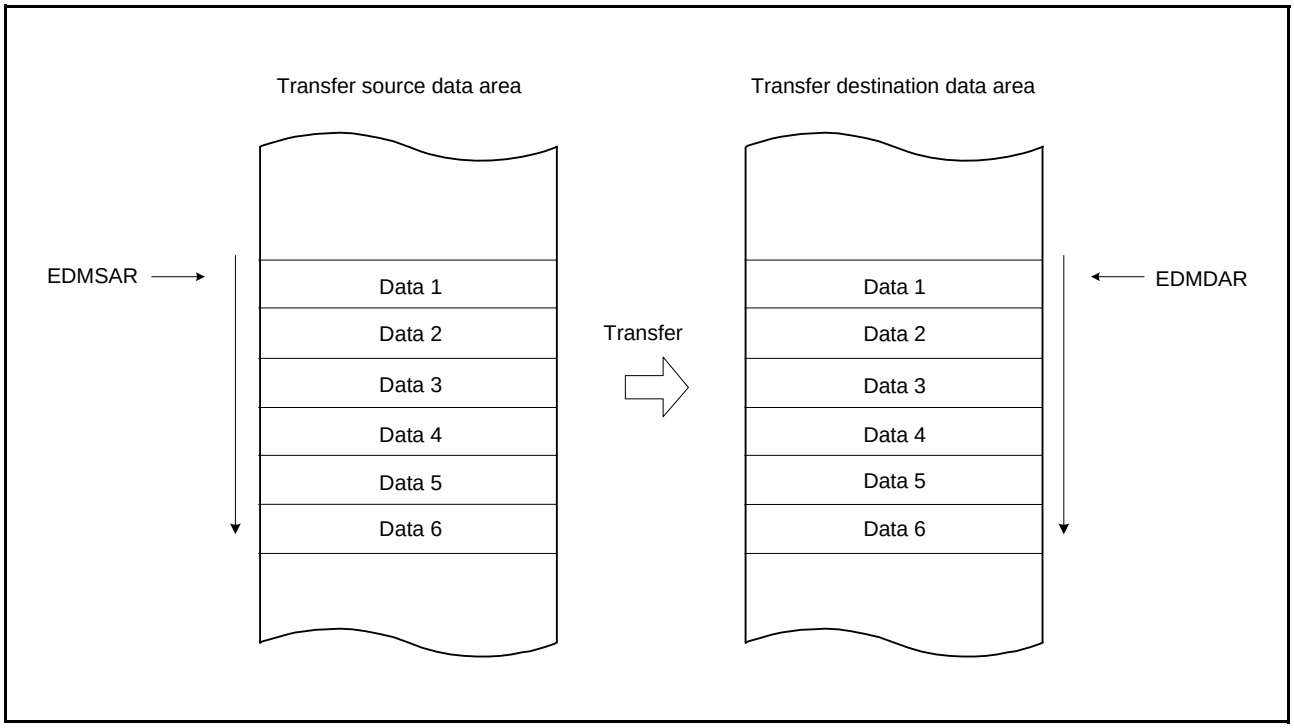


Figure 18.2 Operation in Normal Transfer Mode

(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 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 18.5 summarizes the register update operation in repeat transfer mode, and Figure 18.3 shows the operation in repeat transfer mode.

Table 18.5 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*	• EXDMACn.EDMTMD.DTS[1:0] = 00 Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 01 Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10 Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	Increment/decrement/fixed/ offset addition*	• EXDMACn.EDMTMD.DTS[1:0] = 00 Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01 Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 10 Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAH	Repeat size	Not updated	Not updated
EXDMACn.EDMCRAL	Transfer count	Decremented by one	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Block count	Not updated	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.

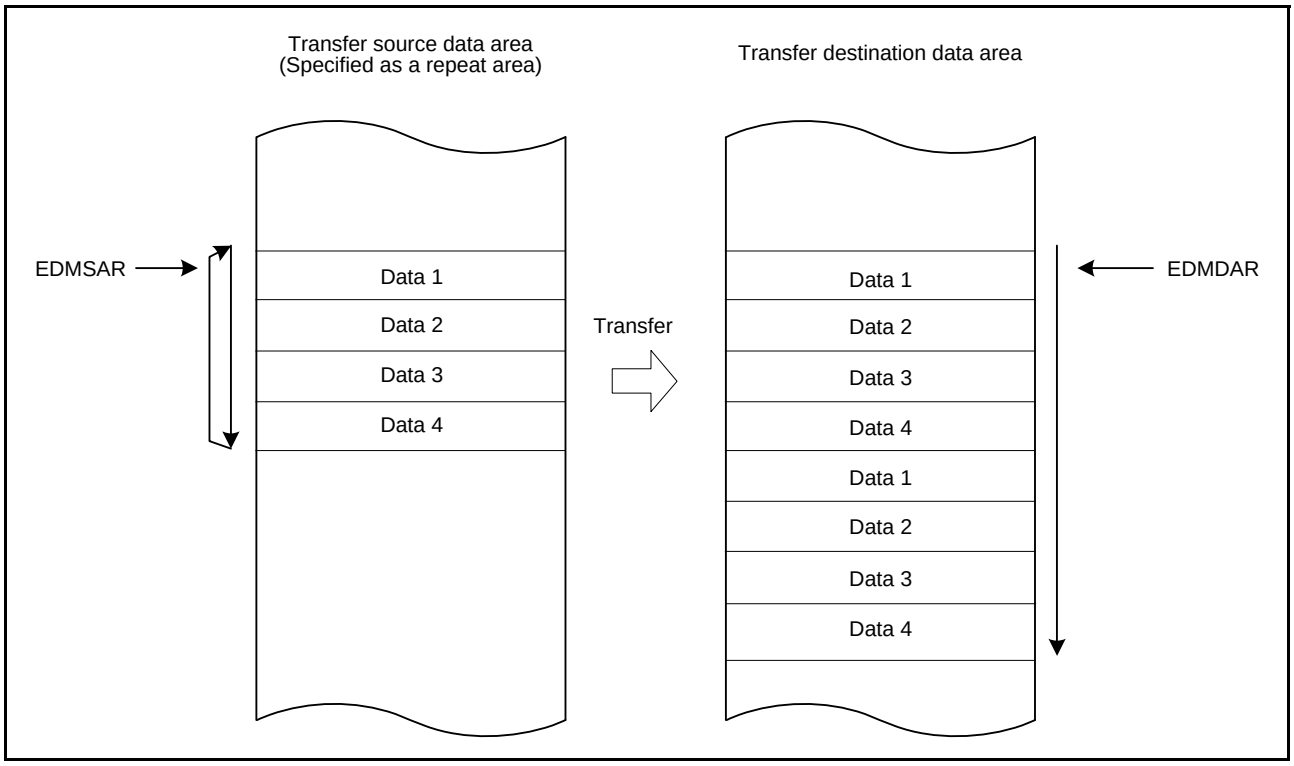


Figure 18.3 Operation in Repeat Transfer Mode

(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 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 18.6 summarizes the register update operation in block transfer mode, and Figure 18.4 shows the operation in block transfer mode.

Table 18.6 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] = 00 Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 01 Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10 Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	• EXDMACn.EDMTMD.DTS[1:0] = 00 Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01 Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 10 Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAH	Block size	Not updated
EXDMACn.EDMCRAL	Transfer count	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Block count	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.

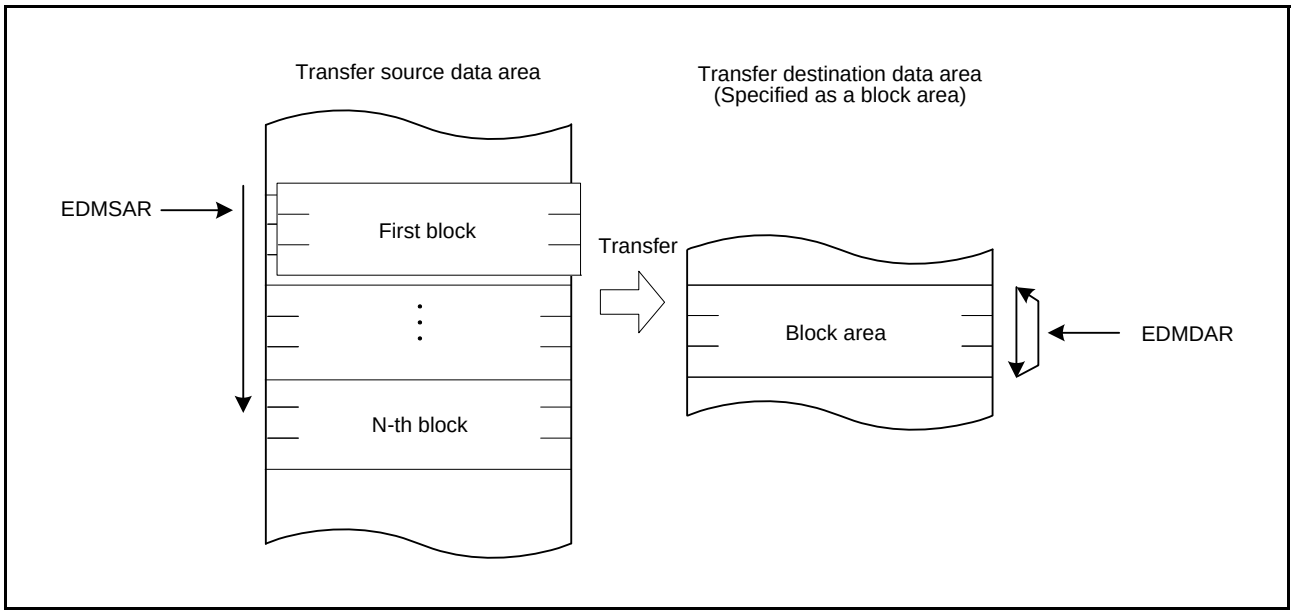


Figure 18.4 Operation in Block Transfer Mode

(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 8K 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] = 11, 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] = 11, 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] = 11, 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 18.7 summarizes the register update operation in cluster transfer mode, and Figure 18.5 shows the operation in cluster transfer mode.

Table 18.7 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	Transfer count	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Cluster count	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.
In read address mode, the transfer destination address EXDMACn.EDMADAR is fixed (invalid).
In write address mode, the transfer destination address EXDMACn.EDMASAR is fixed (invalid).

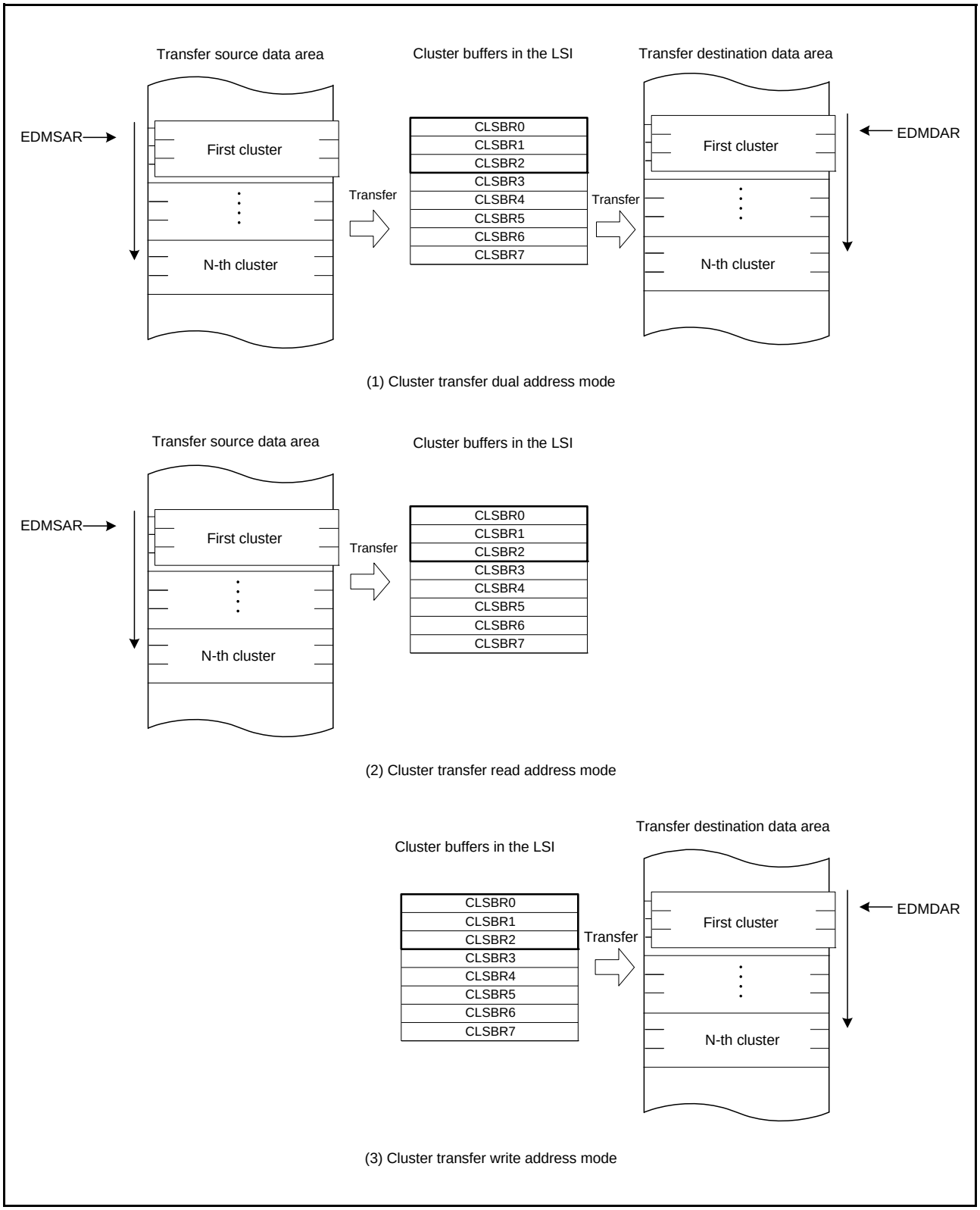


Figure 18.5 Operation in Cluster Transfer Mode

18.3.2 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 cleared 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 18.6 shows an example of the extended repeat area operation.

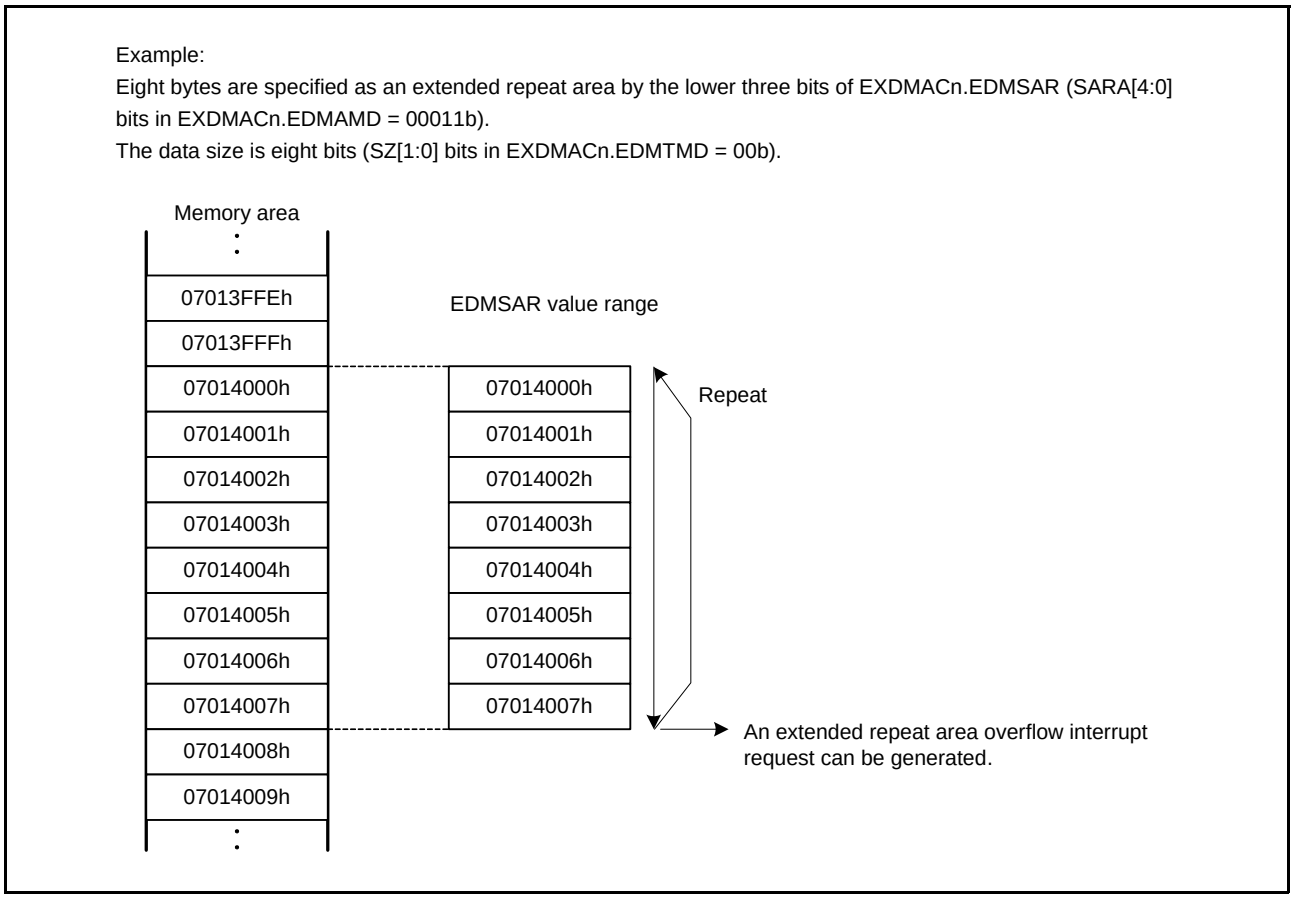


Figure 18.6 Example of Extended Repeat Area Operation

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 18.7 shows an example when the extended repeat area function is used in block transfer mode.

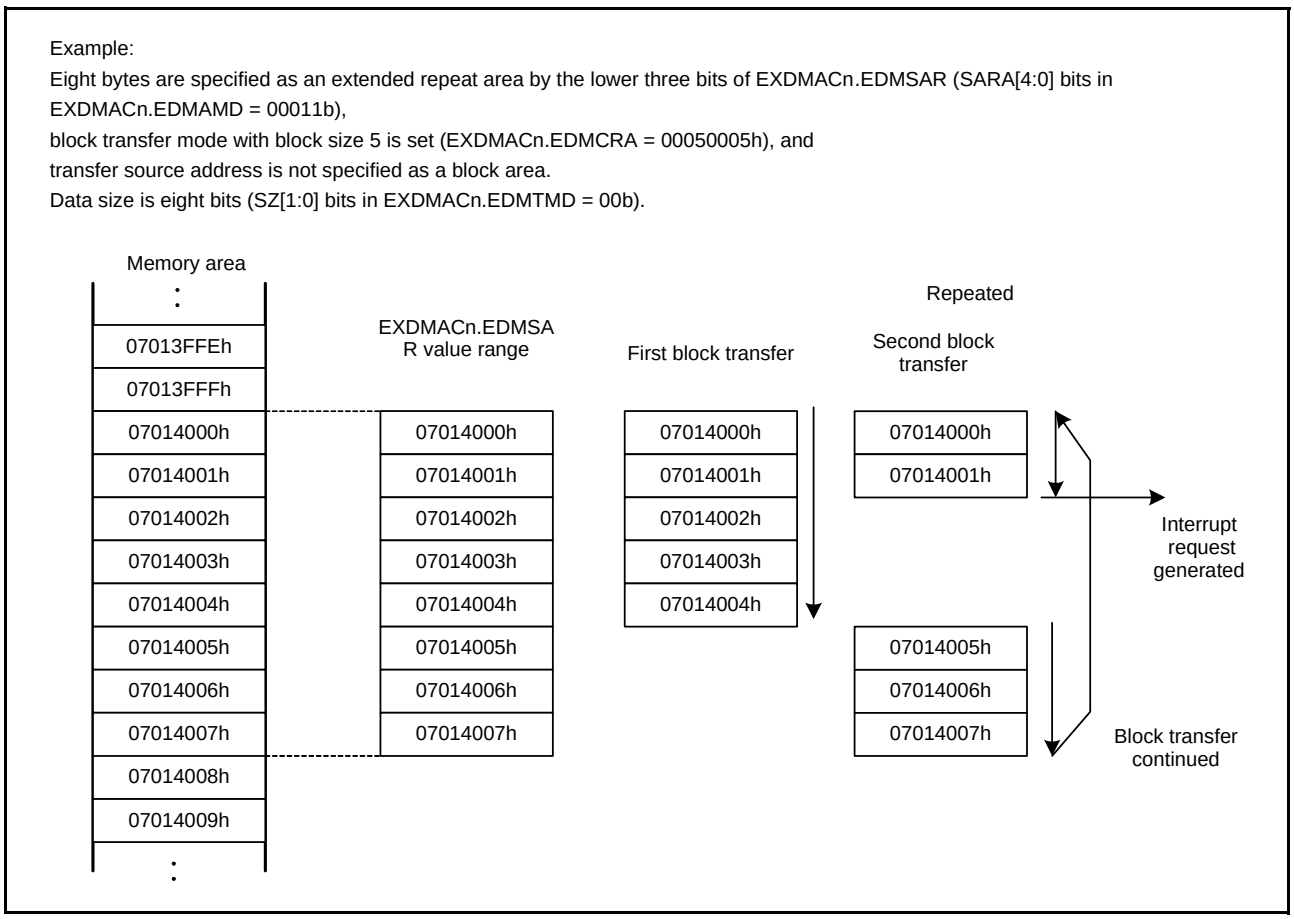


Figure 18.7 Example of Extended Repeat Area Function in Block Transfer Mode

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 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 18.8 shows the address update method in each address update mode.

Table 18.8 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 = 1 + ~offset (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 18.8 shows an example of address updating using offset addition.

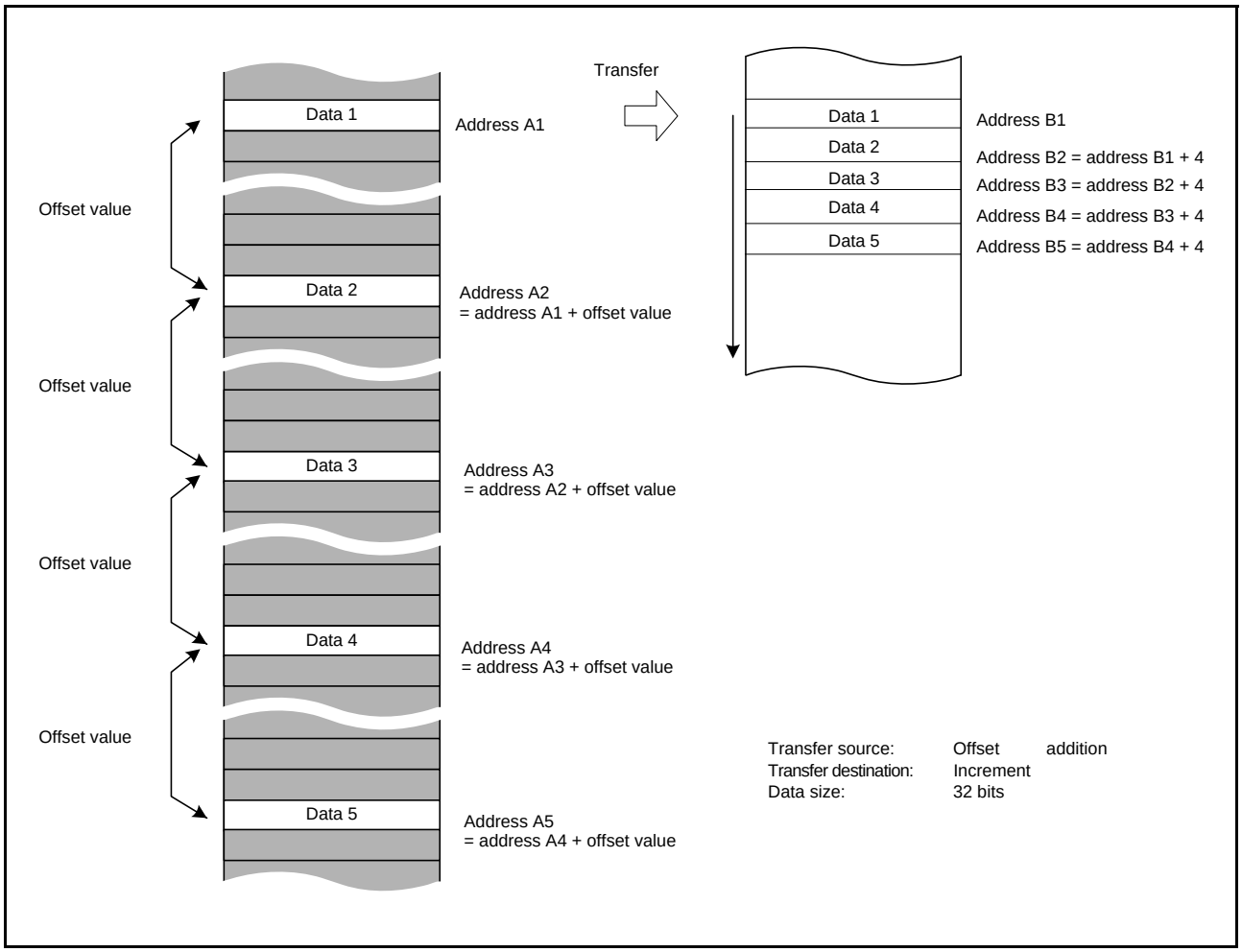


Figure 18.8 Example of Address Updating by Offset Addition

In Figure 18.8, 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 18.9 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

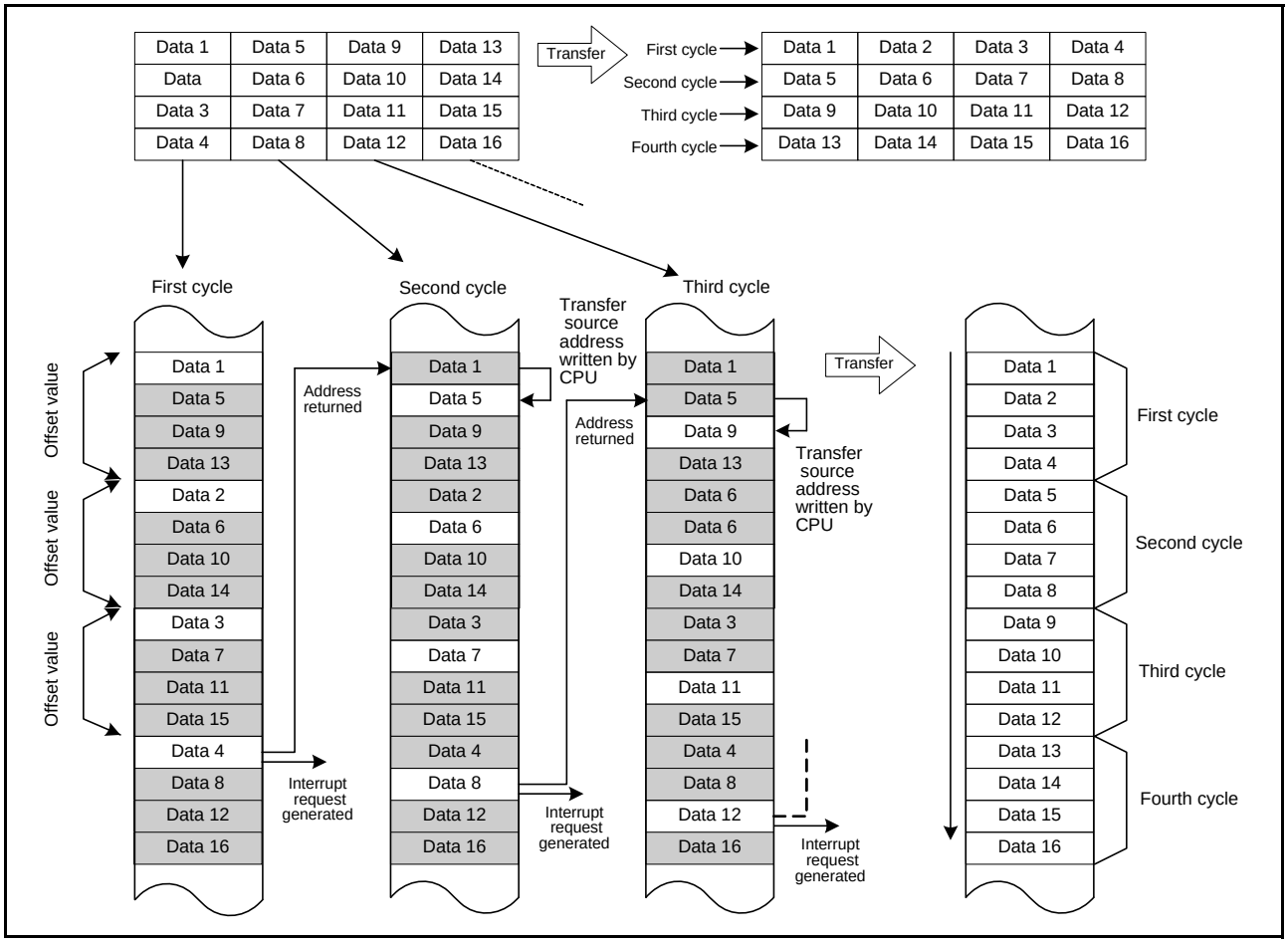


Figure 18.9 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

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, write the address of data 5 to EDMSAR of EXDMAC0 using the CPU (when the data access size is 32 bits, write the data 1 address + 4). When the DTE bit in EDMCNT of EXDMAC0 is set to 1, the transfer is resumed from the state when the 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 18.10 shows a flowchart of the XY conversion.

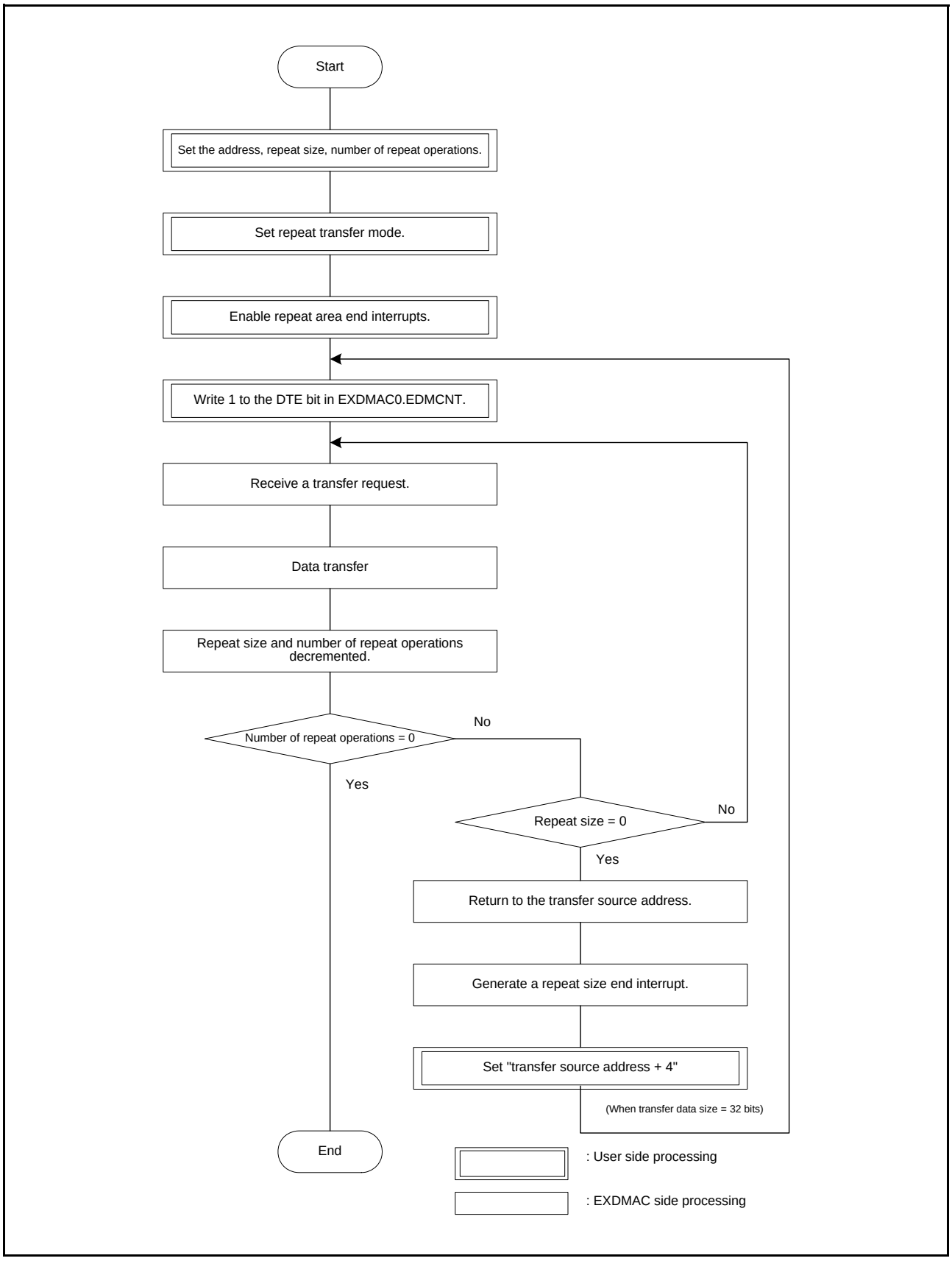


Figure 18.10 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

18.3.4 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 18.9 shows the relationship between transfer modes and address modes.

Table 18.9 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. The RX63N/RX631 receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. The RX63N/RX631 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. The RX63N/RX631 receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. The RX63N/RX631 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 (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. The RX63N/RX631 receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. The RX63N/RX631 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.

18.4 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.

18.4.1 Normal/Repeat Transfer Operation

(1) Dual Address Mode

Figure 18.11 shows the bus cycle example in normal-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, which is started at the falling edge of EDREQ.
The bus cycles in repeat-transfer dual address mode are the same as those in normal-transfer dual address mode.

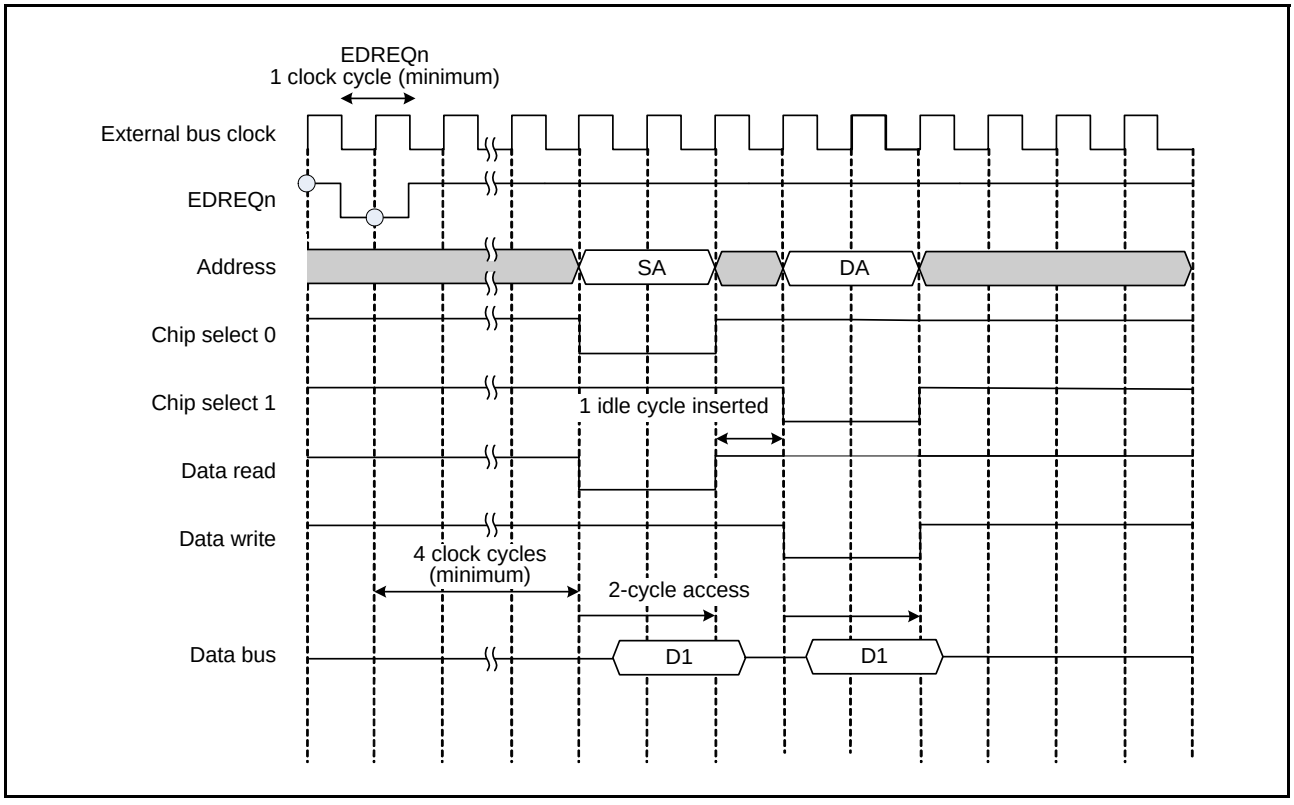


Figure 18.11 Bus Cycle Example in Normal-Transfer Dual Address Mode

(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 18.12 shows the data flow in single address mode.

Figure 18.13 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).

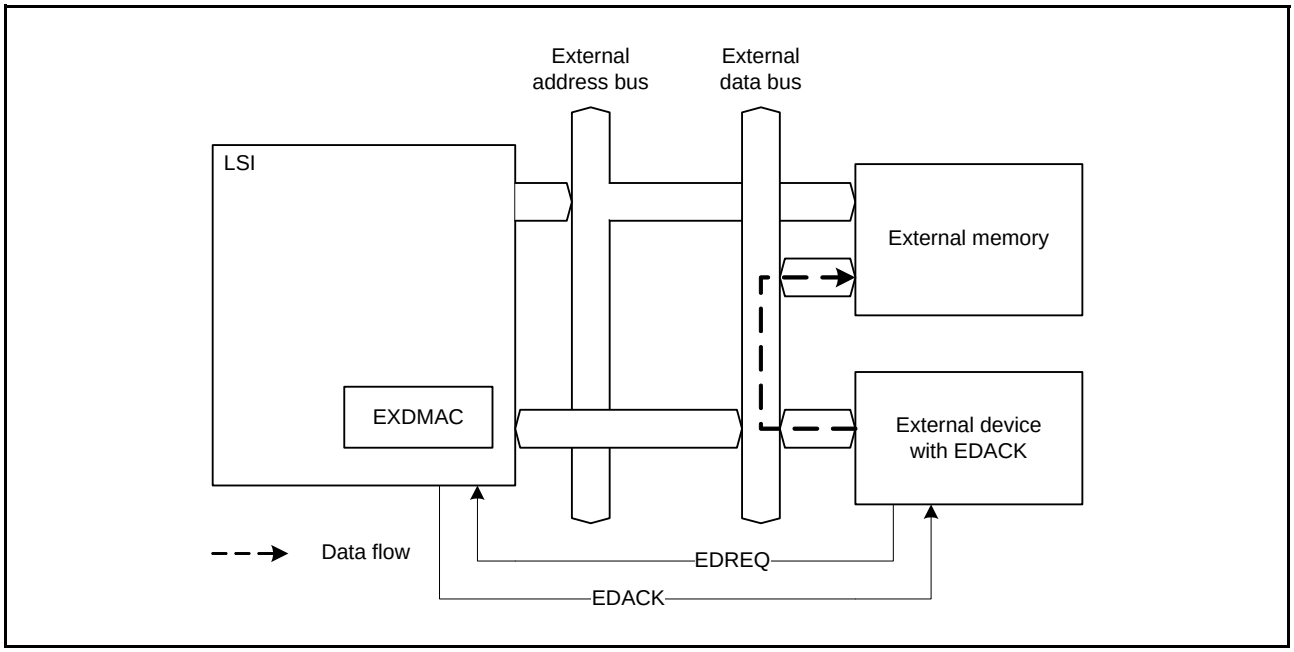


Figure 18.12 Data Flow in Single Address Mode (when EDMAMD.DIR = 1)

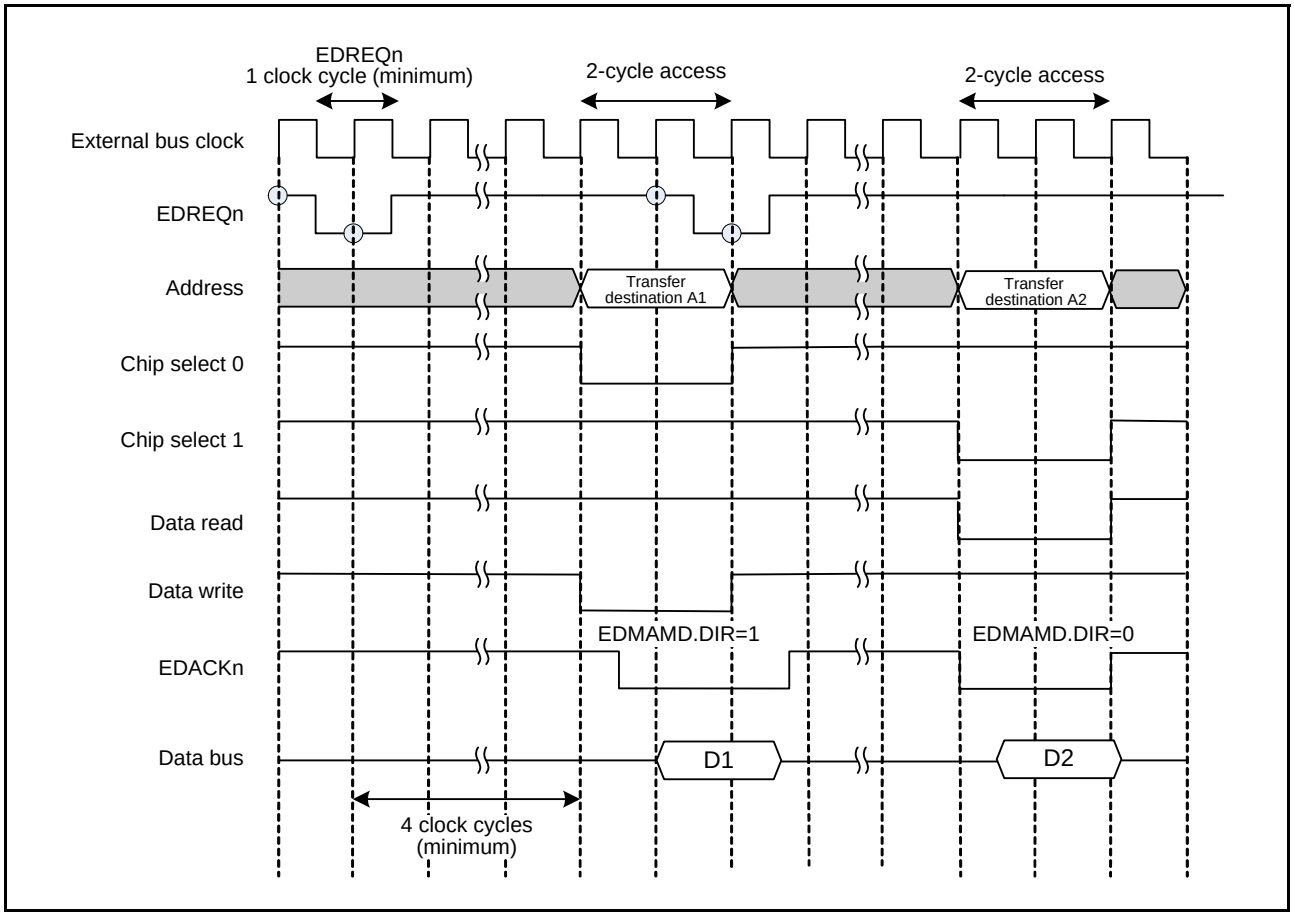


Figure 18.13 Bus Cycle Example in Normal-Transfer Single Address Mode

18.4.2 Block Transfer Operation

(1) Dual Address Mode

Figure 18.14 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.

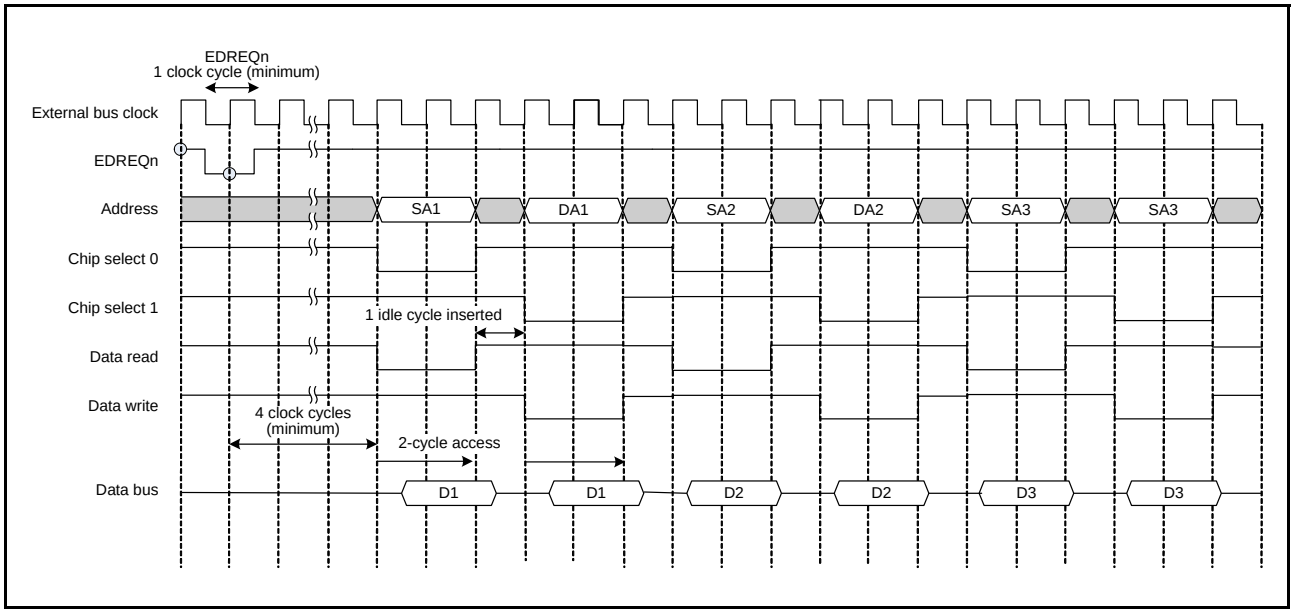


Figure 18.14 Bus Cycle Example in Block-Transfer Dual Address Mode

(2) Single Address Mode

Figure 18.15 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.

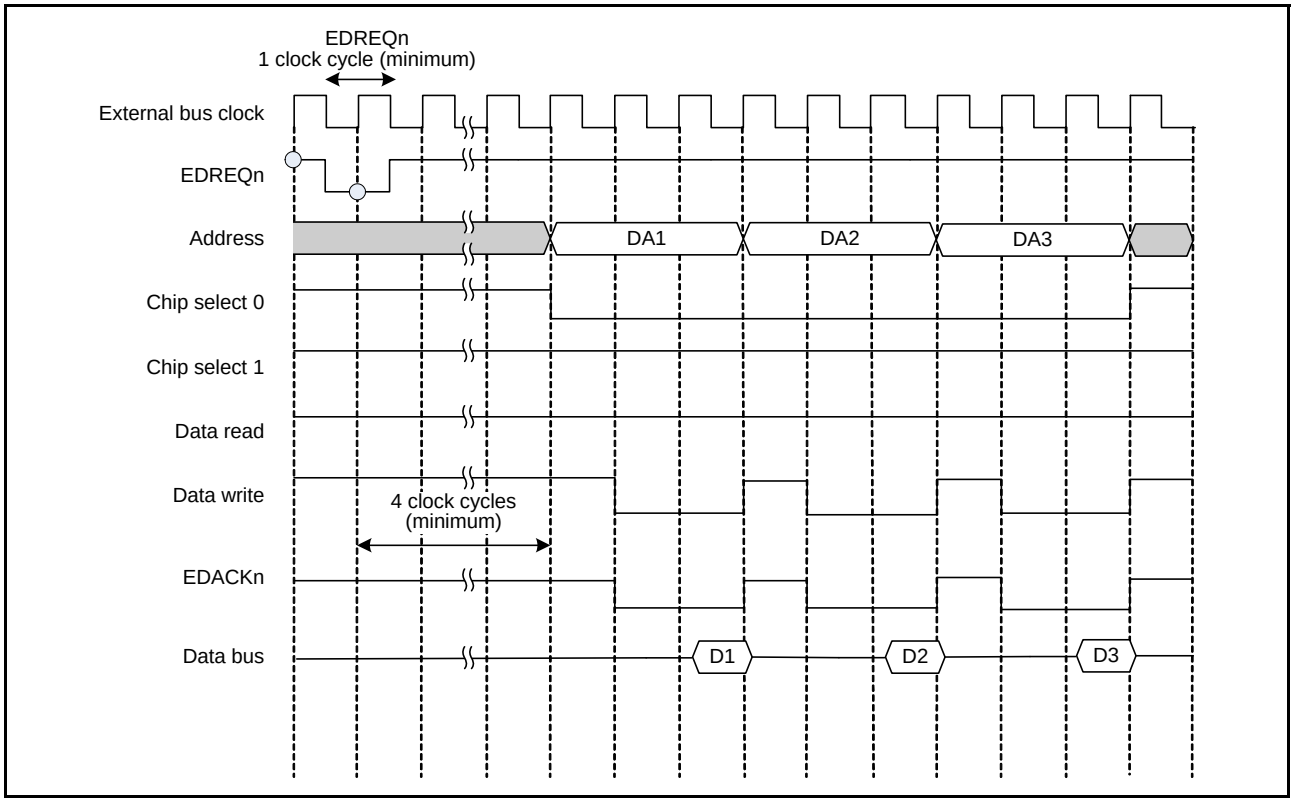


Figure 18.15 Data Flow in Block Transfer Single Address Mode

18.4.3 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 18.16 shows the data flow in cluster-transfer dual address mode and Figure 18.17 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is three.

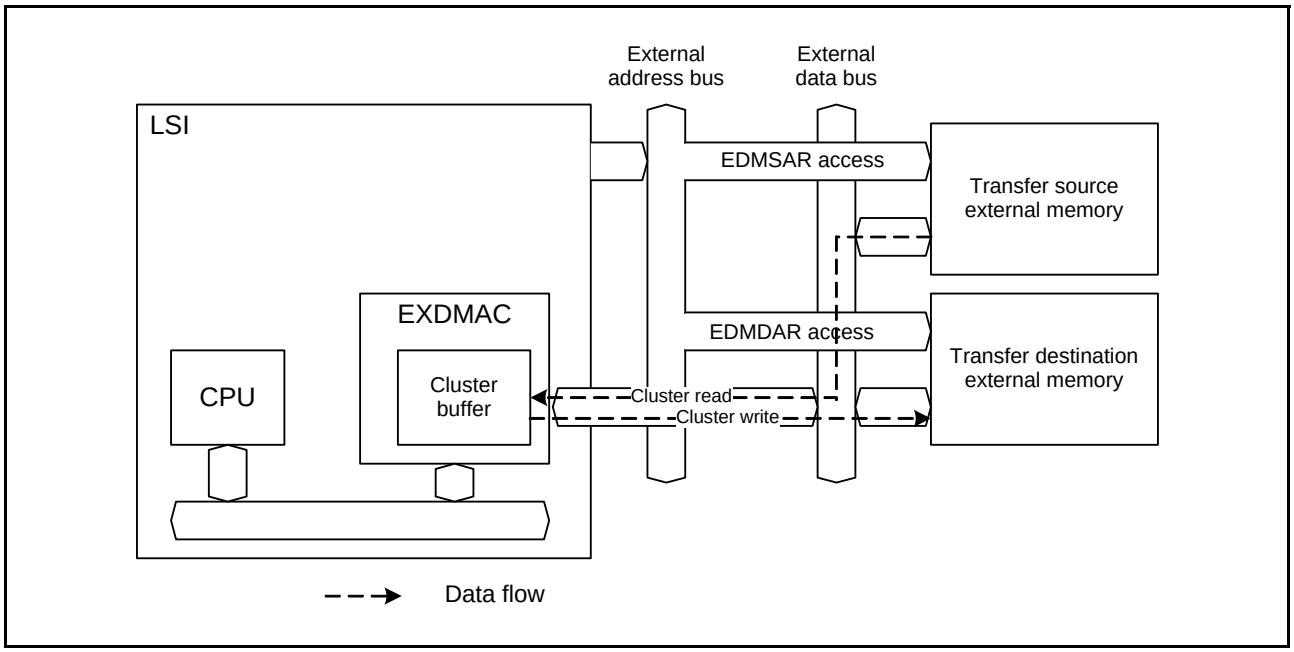


Figure 18.16 Data Flow in Cluster-Transfer Dual Address Mode

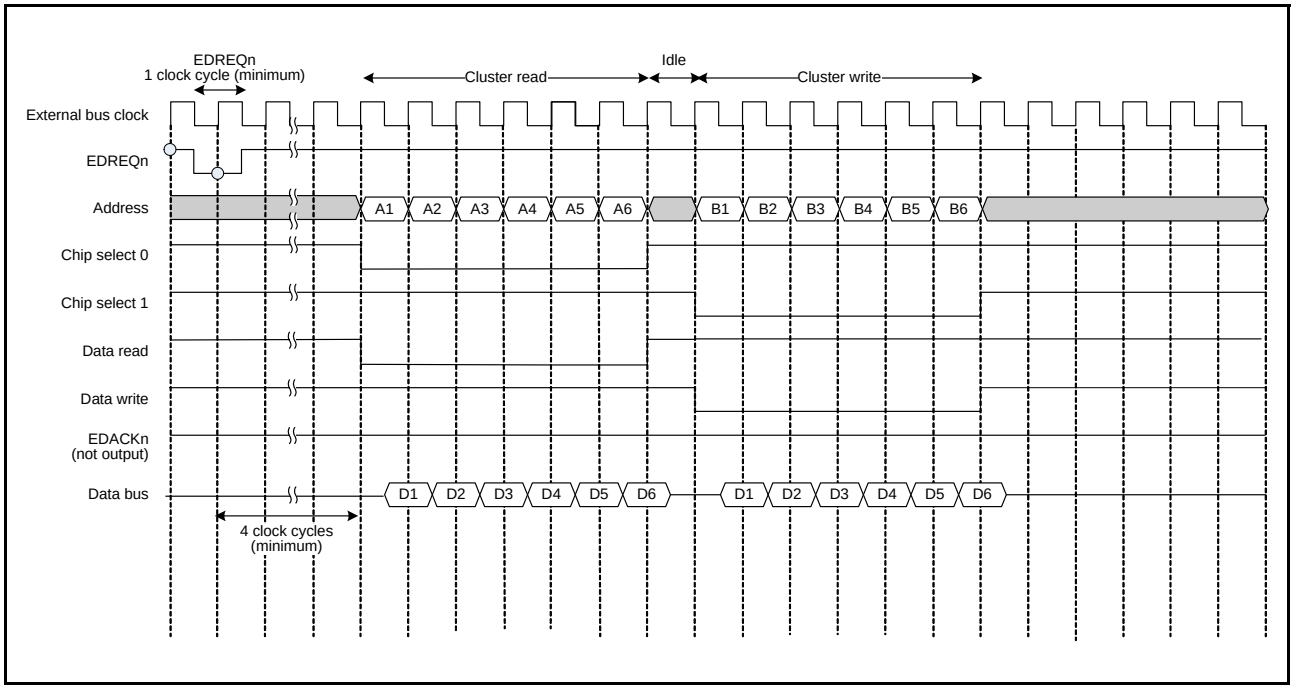


Figure 18.17 Bus Cycle Example in Cluster-Transfer Dual Address Mode

(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 18.18 shows the data flow in cluster-transfer read address mode and Figure 18.19 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is six.

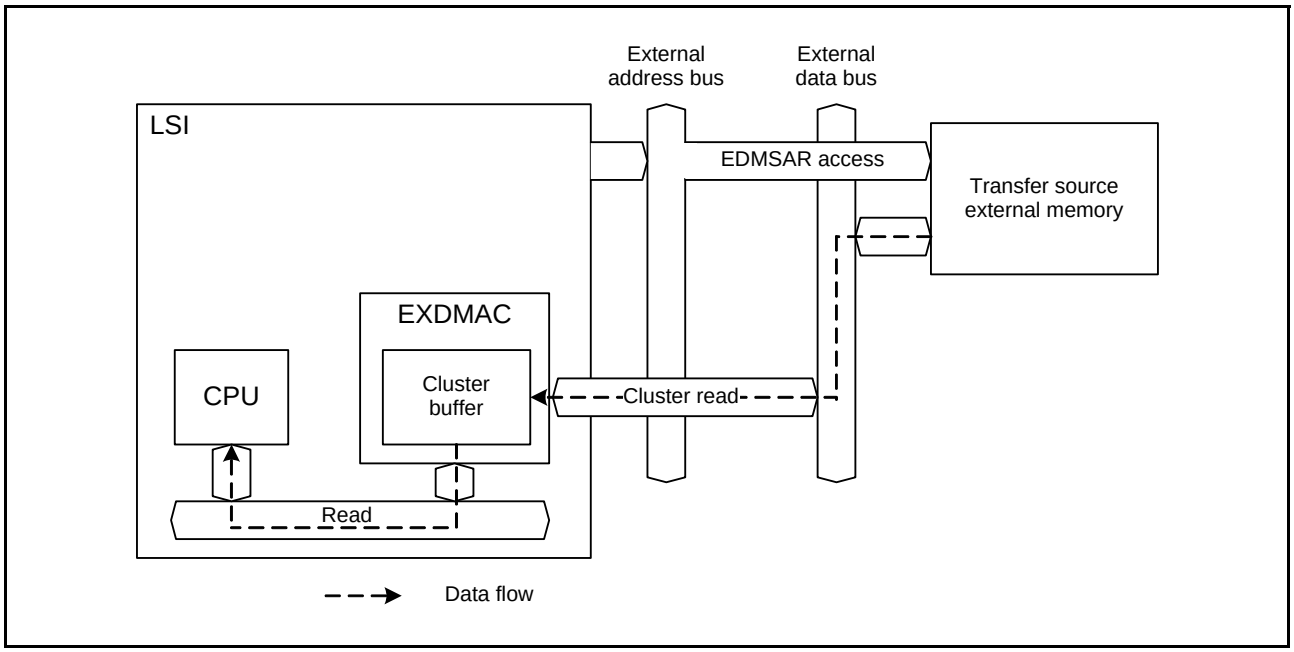


Figure 18.18 Data Flow in Cluster-Transfer Read Address Mode

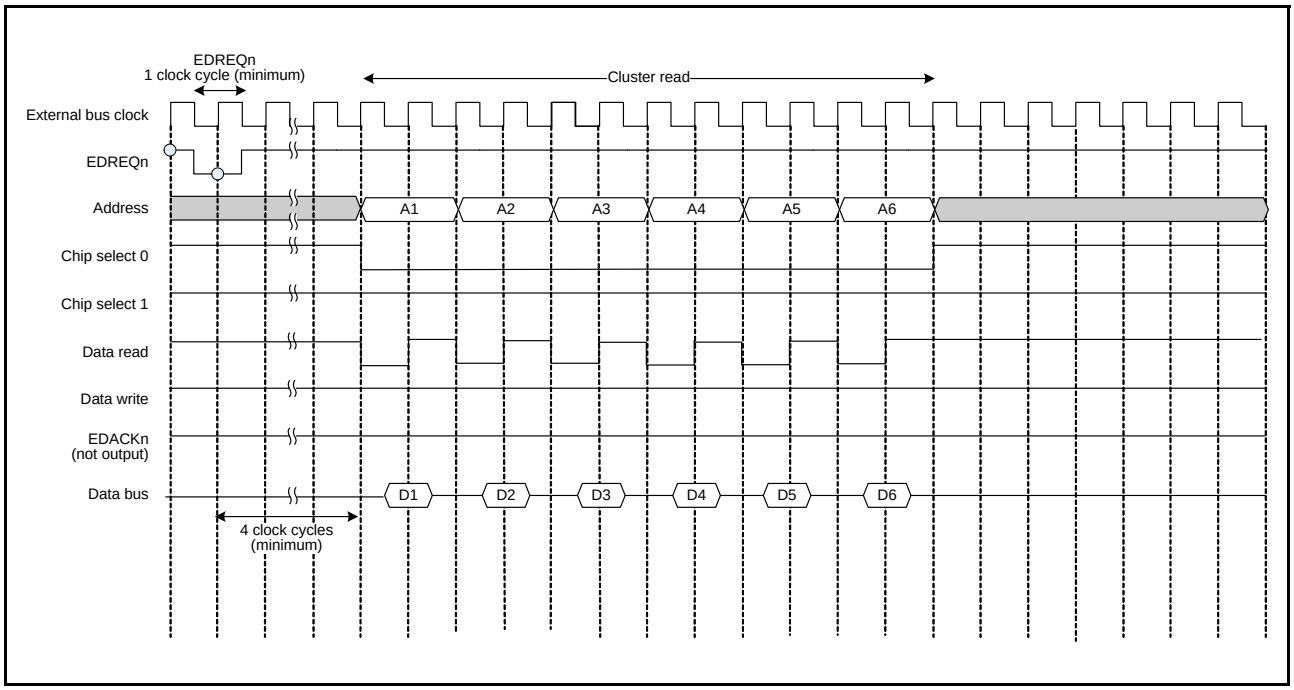


Figure 18.19 Bus Cycle Example in Cluster-Transfer Read Address Mode

(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 18.20 shows the data flow in cluster-transfer write address mode and Figure 18.21 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is six.

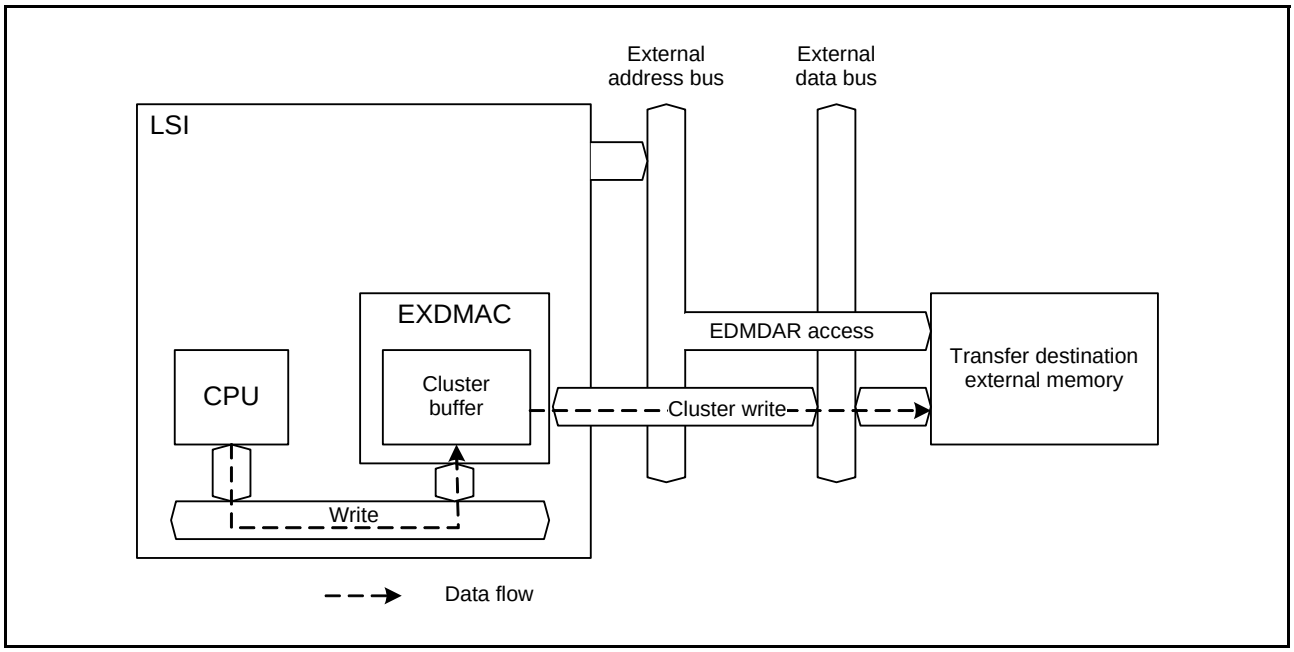


Figure 18.20 Data Flow in Cluster-Transfer Write Address Mode

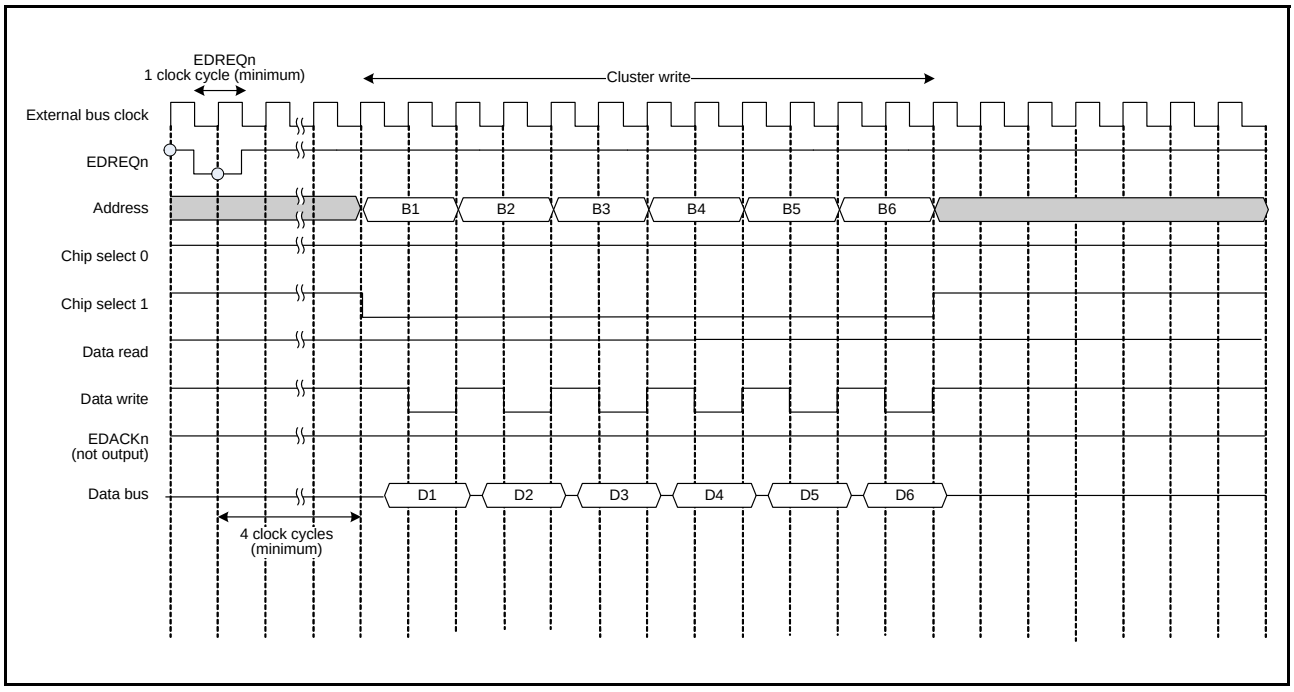


Figure 18.21 Bus Cycle Example in Cluster-Transfer Write Address Mode

18.5 Activation Sources and Procedures

18.5.1 Activation Sources

The EXDMAC can be activated by software, by an external DMA transfer request pin (EDREQn pins), or by the DMA transfer requests from the peripheral modules (compare match A of MTU1 or TPU7). Setting the DCTG[1:0] bits in EDMTMD of EXDMACn selects the activation source.

(1) EXDMAC Activation by Software

Setting the EXDMACn.EDMREQ.DCTG[1:0] bits to 00b enables the EXDMAC activation 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 cleared 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 cleared to 0. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) EXDMAC Activation by external DMA transfer request Pin (EDREQn)

Setting the EXDMACn.EDMTMD.DCTG[1:0] bits to 10b enables the EXDMAC activation by the external DMA transfer request pins.
To set the activation 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 (activation enabled).

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 cleared to 0 when the DMA transfer is started by the external request. Moreover, this flag is cleared 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 cleared 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 (EXDMAC activation enable), 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 18.22 and Figure 18.23 show the external DMA transfer request timings in the falling edge detection mode and low level detection mode, respectively.

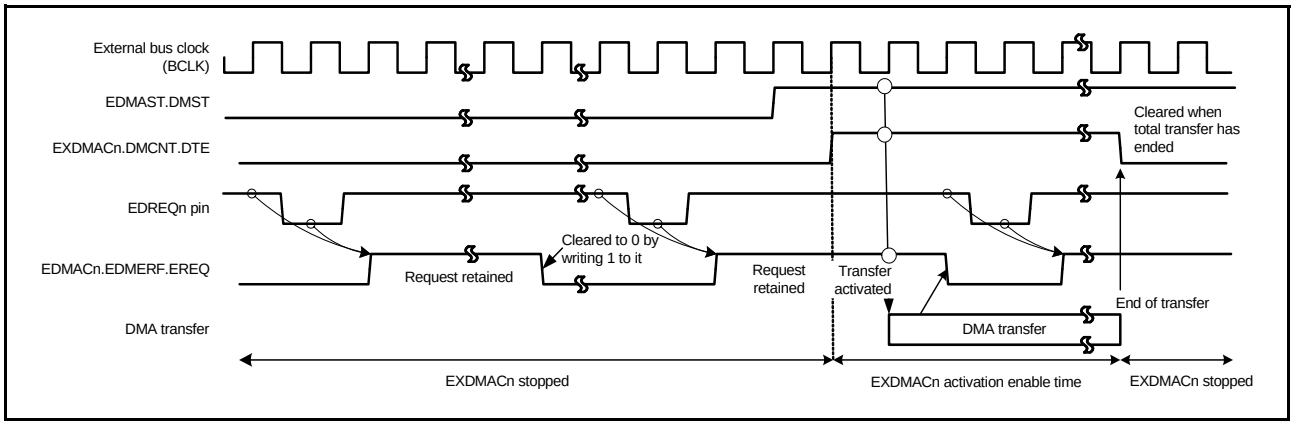


Figure 18.22 External DMA Transfer Request Timing in Falling-Edge Detection Mode

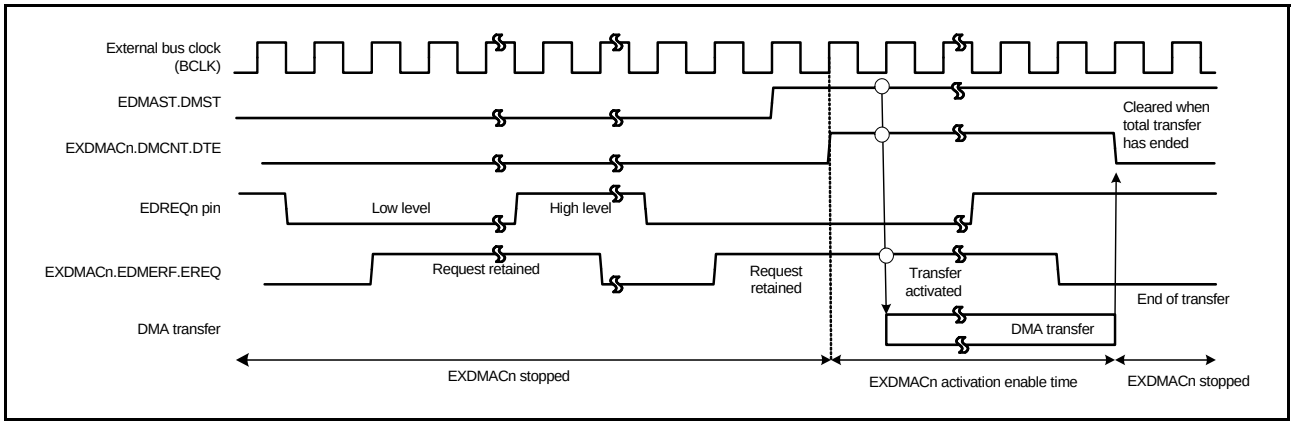


Figure 18.23 External DMA Transfer Request Timing in Low-Level Detection Mode

(3) EXDMAC Activation by DMA Transfer Requests from Peripheral Modules (Compare Match of MTU1 or TPU7)

Setting the DCTG[1:0] bits in EDMTMD of EXDMACn to 11b enables the EXDMAC activation by DMA transfer requests from the peripheral modules (compare match A of MTU1 or TPU7).
To start DMA transfer by DMA transfer requests from the peripheral modules follow the procedure below.

1. Set the unit select bit (SEL.CNj) (For the setting of unit select bit, follow the procedure described in section 15.5.1.5, Unit Selection and Interrupt Status Flags).
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.
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 activations 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 cleared to 0 when the DMA transfer is started by the peripheral module request (DMA transfer is not requested).
This flag is cleared to 0 by writing 1 to it.
When the EDMAST.DMST bit is set to 1 (EXDMAC activation enable) 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.

18.5.2 Activating the EXDMAC

Figure 18.24 shows the register setting procedure.

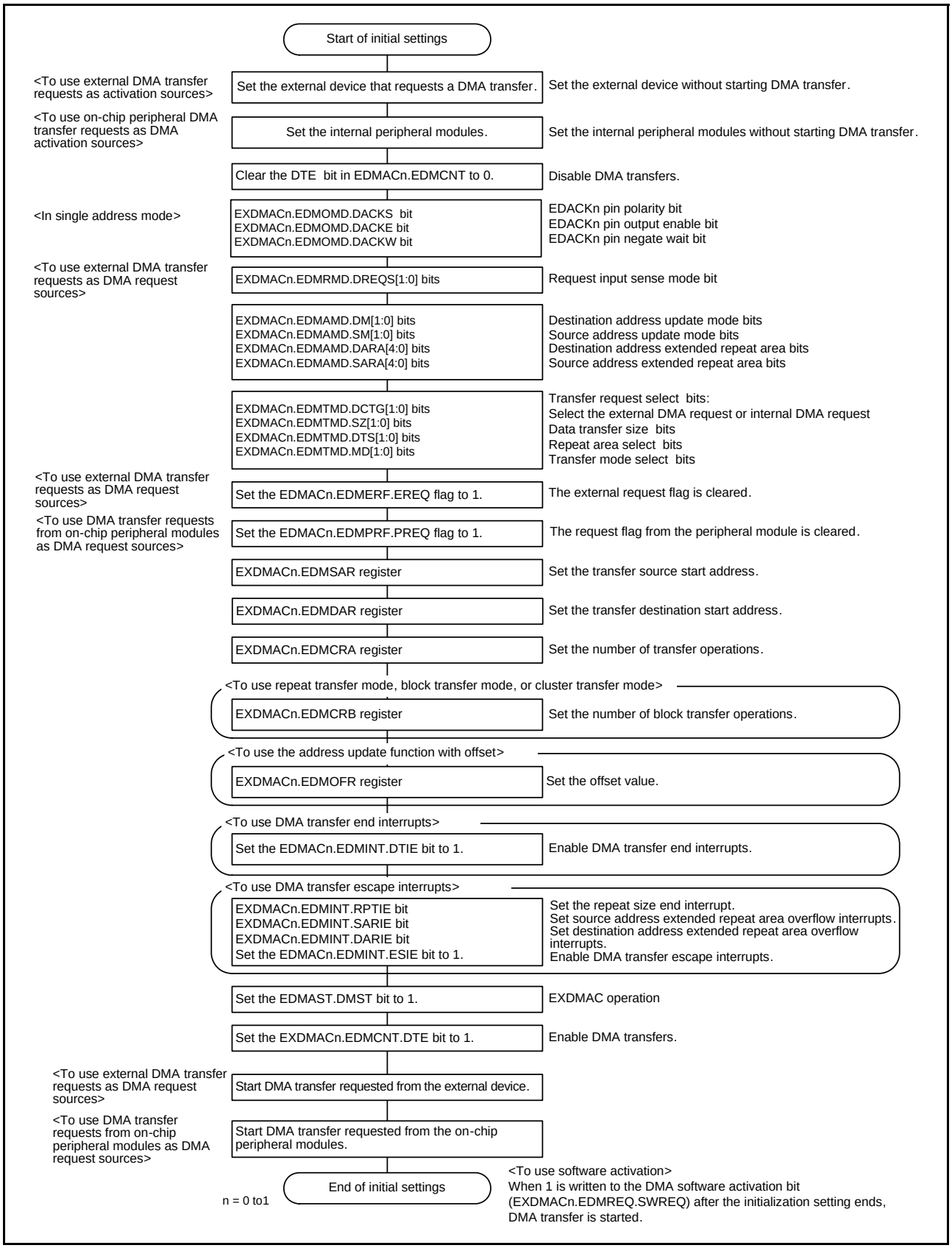


Figure 18.24 Register Setting Procedure

18.5.3 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).

18.5.4 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 18.4 to Table 18.7.

(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 18.4 to Table 18.7.

(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 18.4 to Table 18.7.

(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 18.4 to Table 18.7.

(5) EXDMA Transfer Enable Bit (EXDMACn.EDMCNT.DTE)

The EXDMACn.EDMCNT.DTE bits provide a way to control enabling or prohibition of transfer by writing to the relevant registers.
The EXDMACn.EDMCNT.DTE bit is cleared 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 cleared 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 bit 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 cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared 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 cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared 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 (ICUb).

18.5.5 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.

18.6 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.

18.6.1 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 cleared 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 cleared 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 (ICUb).

(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 cleared 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 (ICUb).

(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 cleared 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 (ICUb).

18.6.2 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 cleared 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 (ICUb).

18.6.3 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 cleared 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 (ICUb).

18.7 Interrupts

The EXDMAC can output one interrupt request to the CPU or the DTC for each channel. Table 18.10 shows the relation among the interrupt sources, the interrupt status bits, and the interrupt enable bits. Figure 18.25 shows the schematic logic diagram of interrupt outputs. The procedures for suspending or resuming DMA transfer by the EXDMAC interrupt are shown in Figure 18.26.

Table 18.10 Relation among Interrupt Sources, Interrupt Status Bits, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Bits	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		

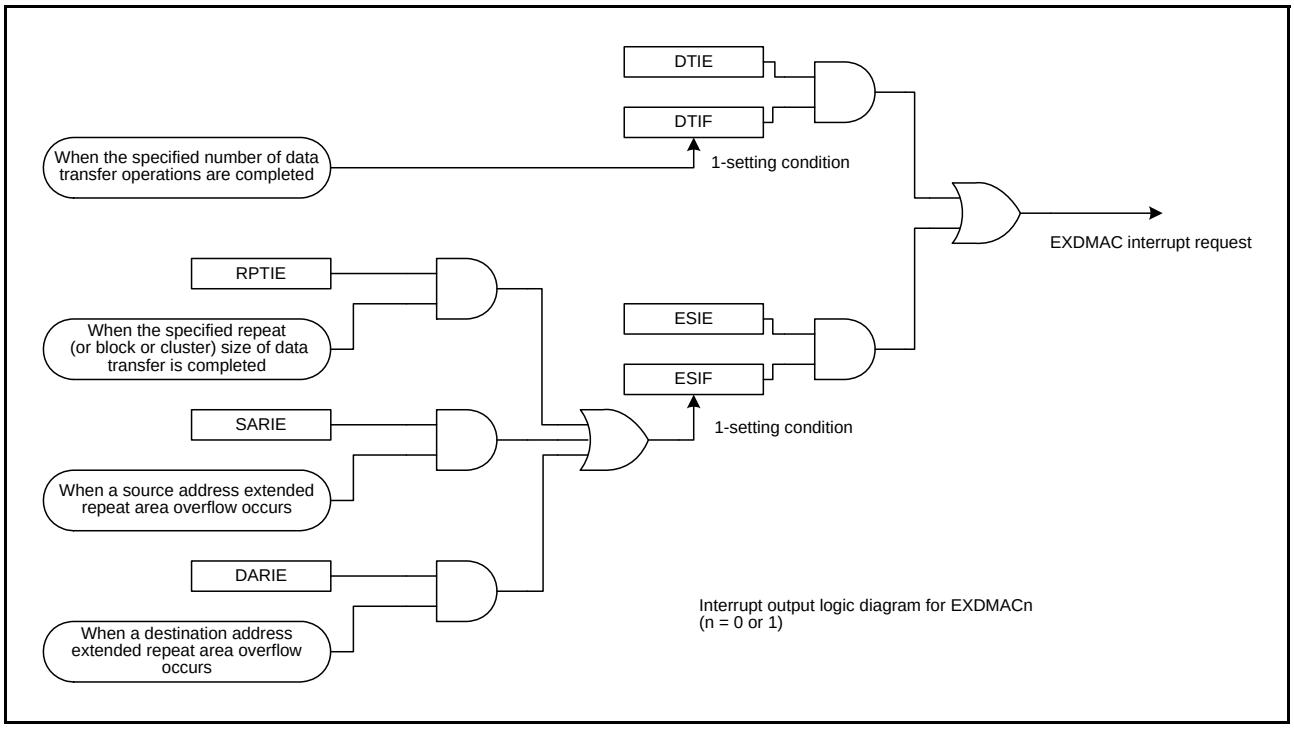


Figure 18.25 Schematic Logic Diagram of Interrupt Outputs

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 bit in EDMSTS of EXDMACn to clear a transfer end interrupt, and to the ESIF bit 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 bit in EDMSTS of EXDMACn is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

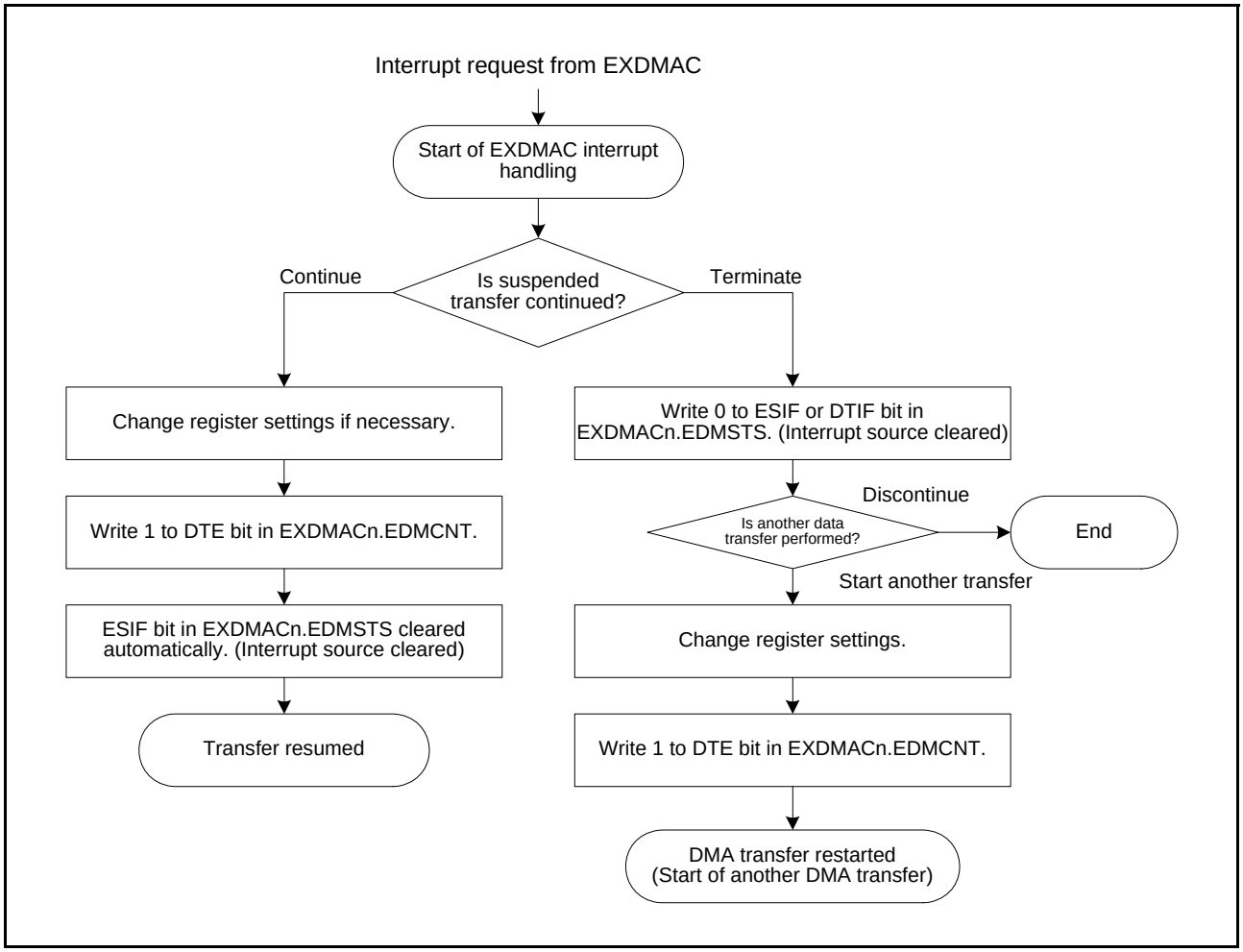


Figure 18.26 Procedures for Suspending or Resuming DMA Transfer by the EXDMAC Interrupt

18.8 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. 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.
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.
To perform DMA transfer after returning from low-power consumption mode, set the EDMAST.DMST bit to 1 again.

18.9 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 EDMOND 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.

18.9.1 EDACKn Operation Example in Normal-Transfer (CS Area) Single Address Mode

Figure 18.27 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 18.28 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.

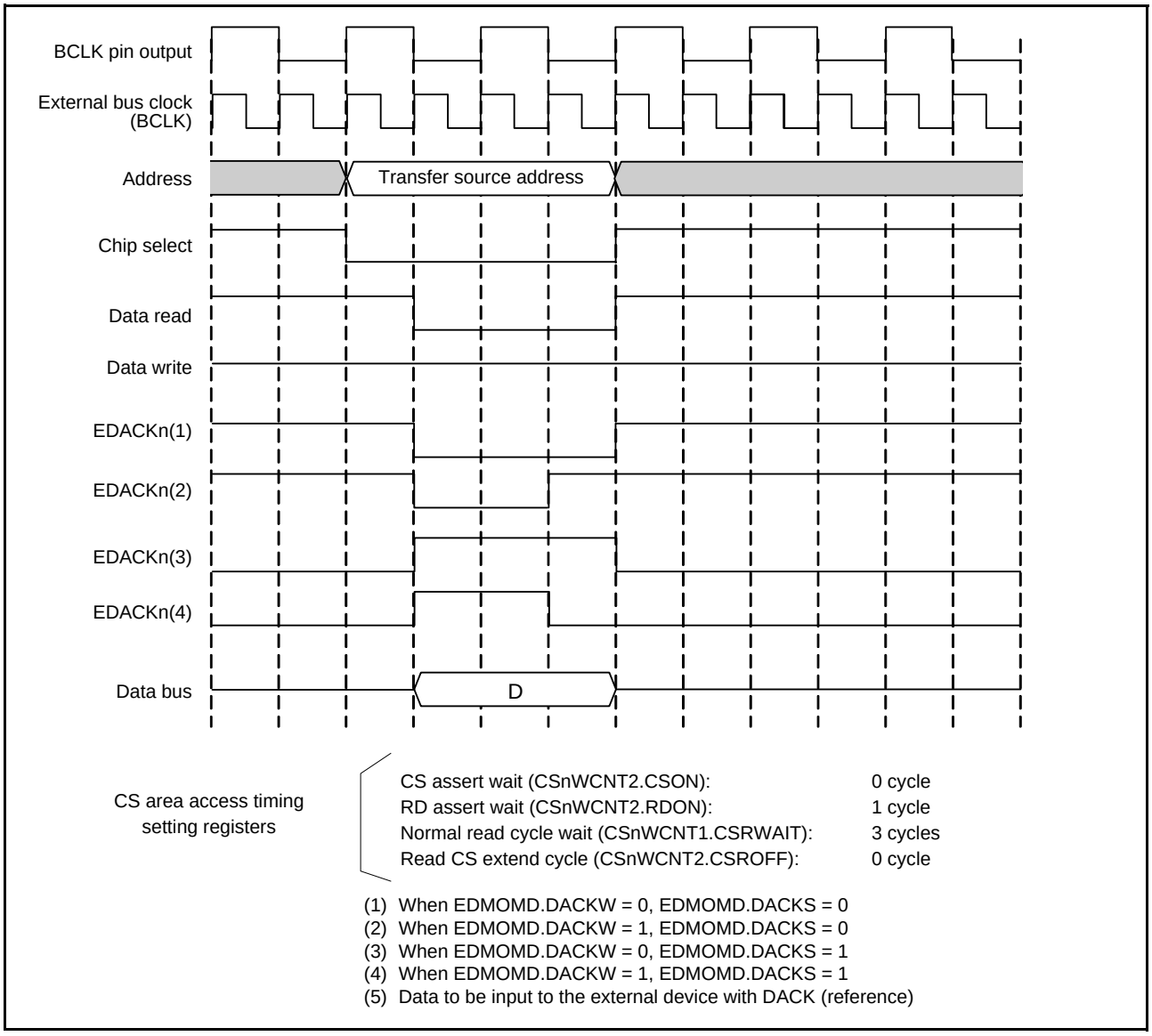


Figure 18.27 Operation Example in Normal-Transfer (CS Area Read) Single Address Mode

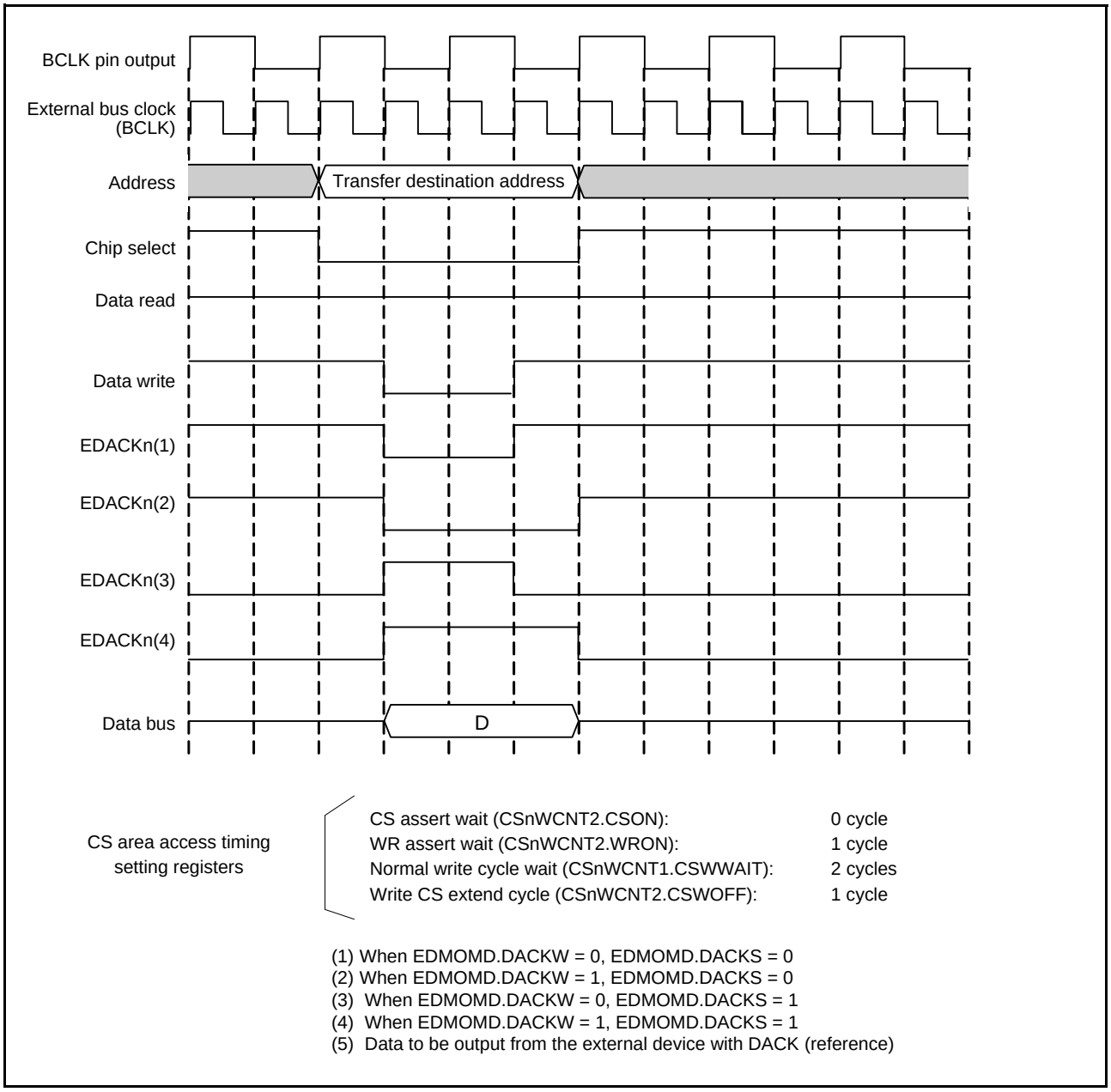


Figure 18.28 Operation Example in Normal-Transfer (CS Area Write) Single Address Mode

18.9.2 EDACKn Operation Example in Normal-Transfer (SDRAM Area) Single Address Mode

Figure 18.29 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 18.30 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.

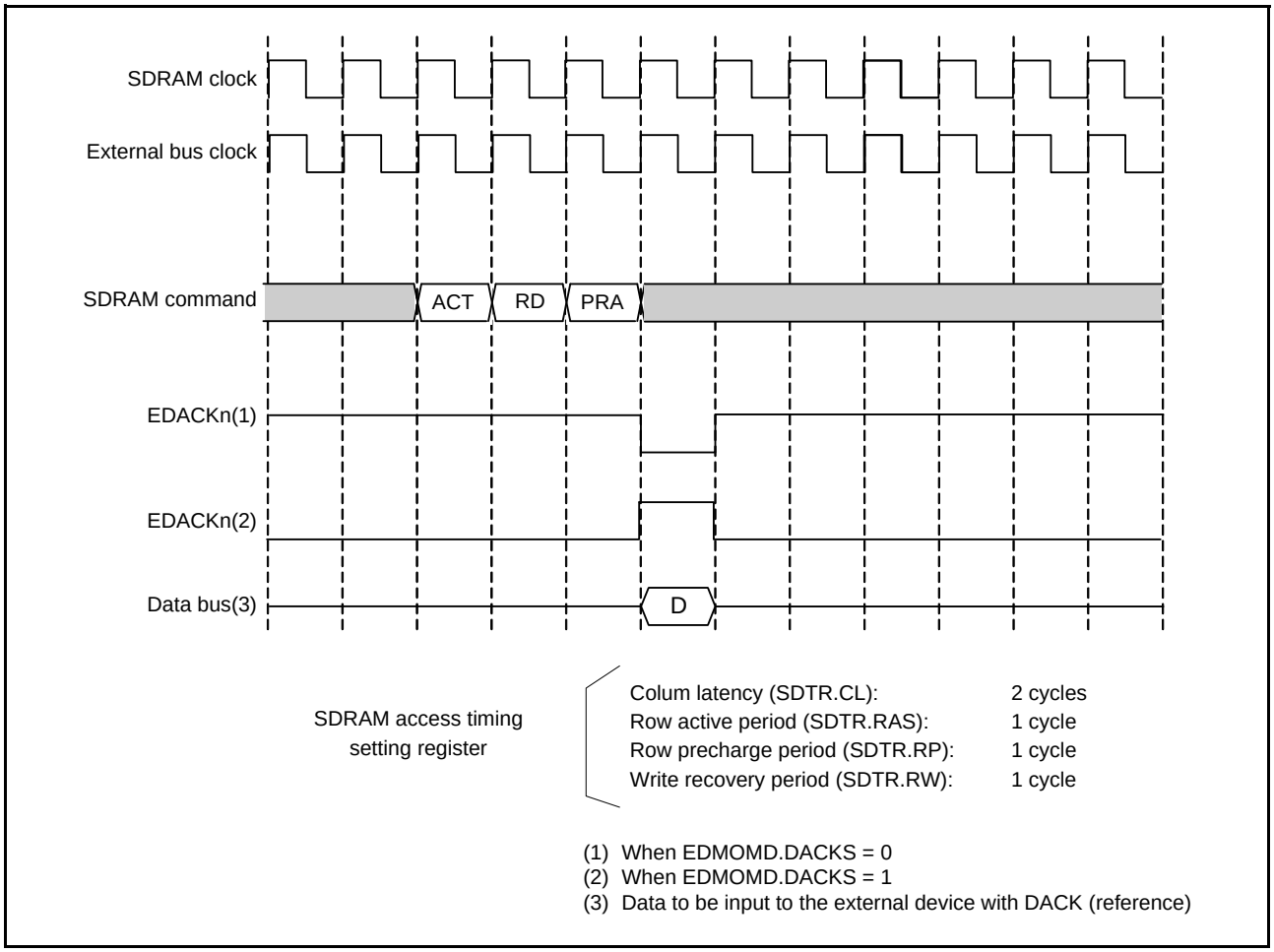


Figure 18.29 Operation Example in Normal-Transfer (SDRAM Area Read) Single Address Mode

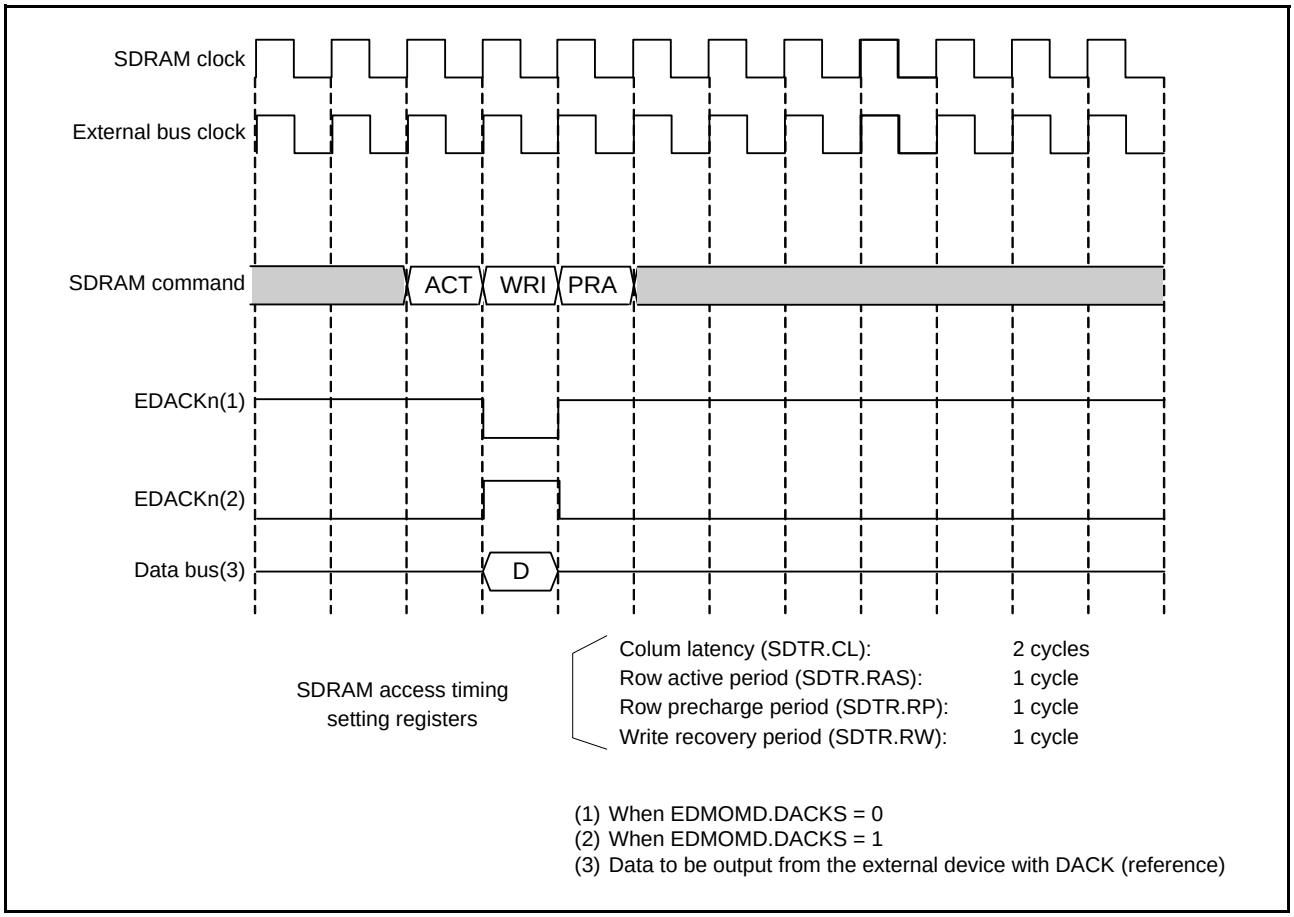


Figure 18.30 Operation Example in Normal-Transfer (SDRAM Area write) Single Address Mode

18.9.3 EDACKn Operation Example in Block-Transfer (CS Area) Single Address Mode

Figure 18.31 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 18.32 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.

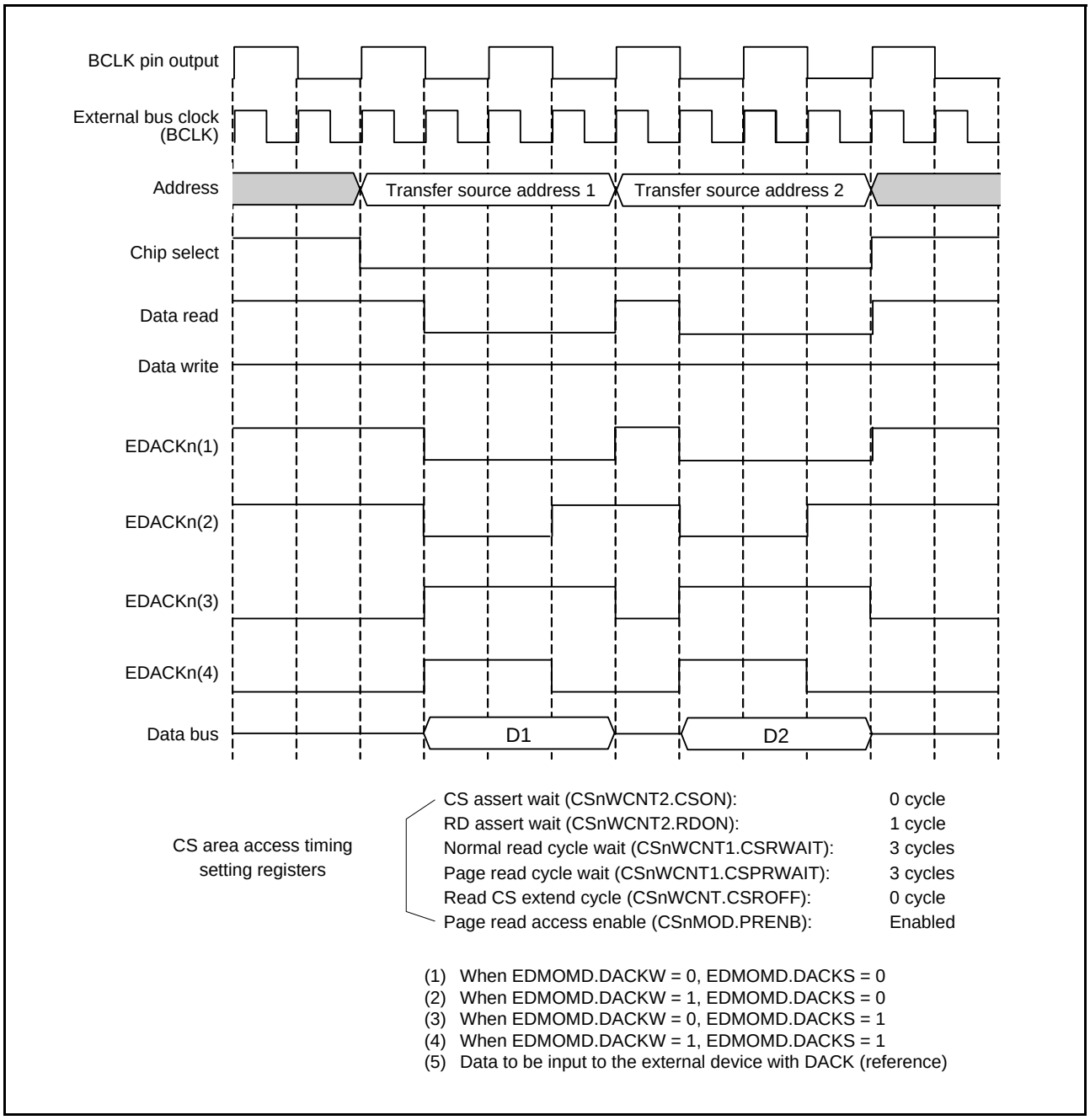


Figure 18.31 Operation Example in Block-Transfer (CS Area Read) Single Address Mode

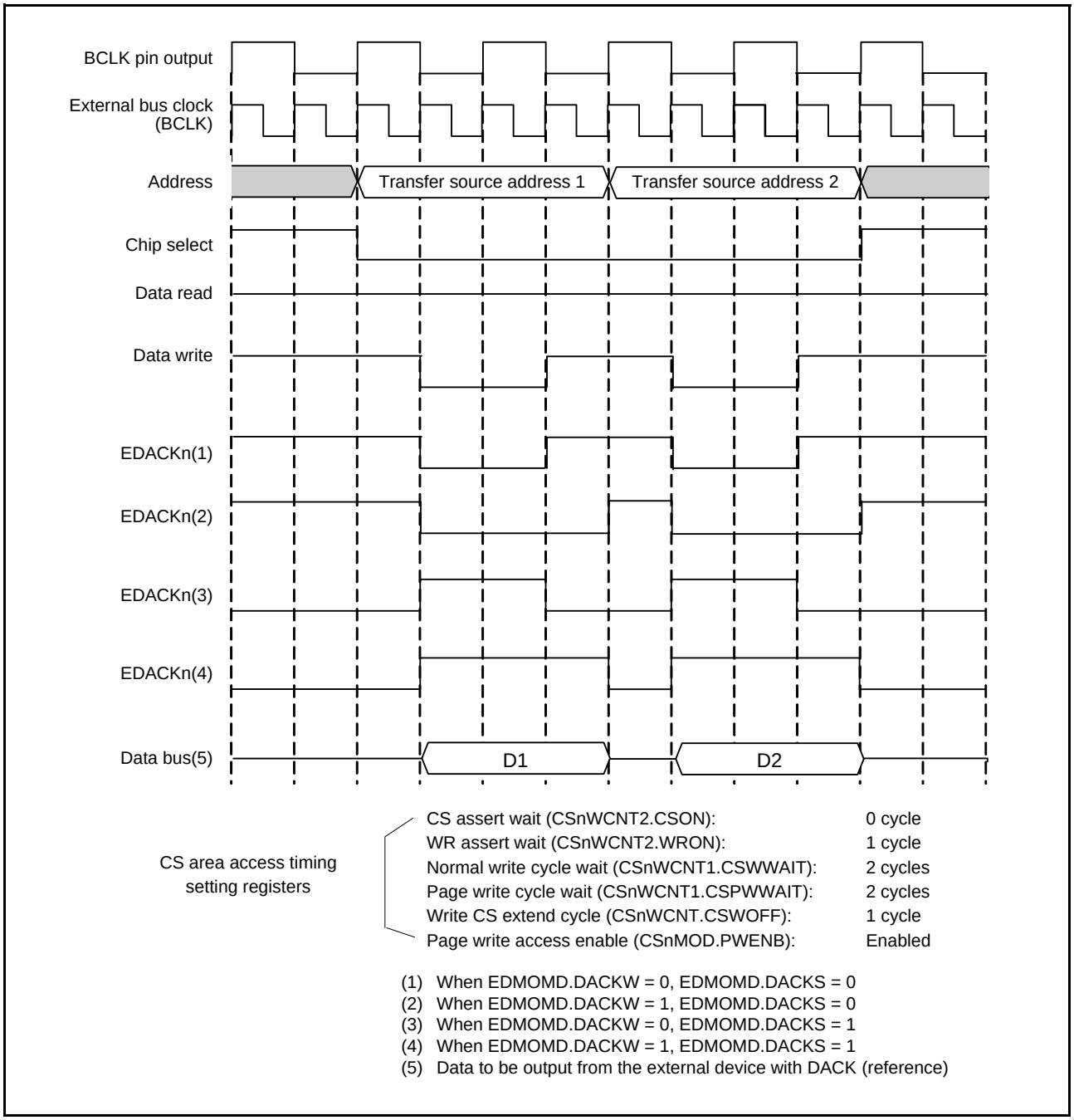


Figure 18.32 Operation Example in Block-Transfer (CS Area Write) Single Address Mode

18.9.4 EDACKn Operation Example in Block-Transfer (SDRAM Area) Single Address Mode

Figure 18.33 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 18.34 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 18.35 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 18.36 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 18.37 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 18.38 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.

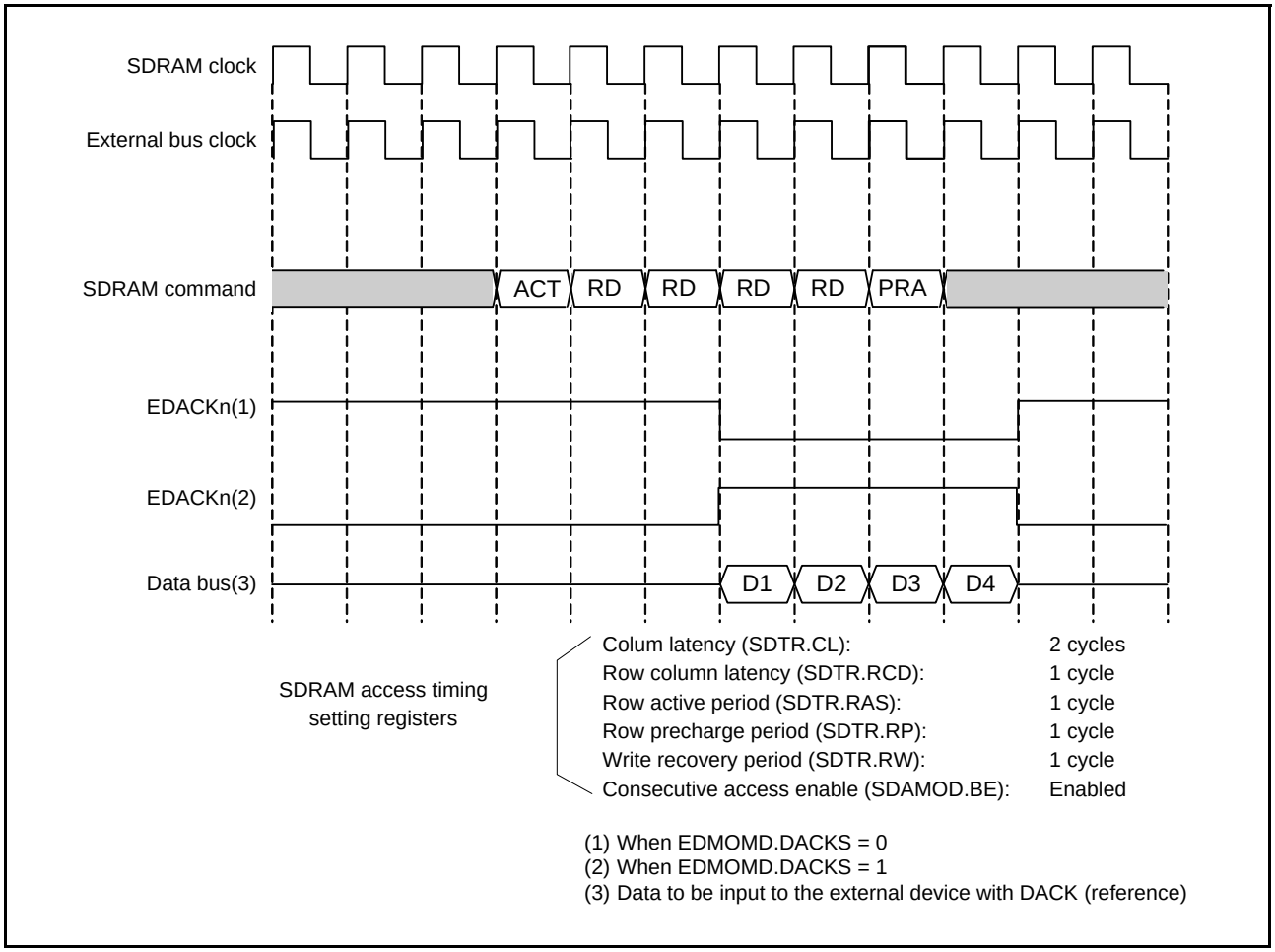


Figure 18.33 Operation Example in Block-Transfer (SDRAM Area Read: Consecutive Access Enabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode

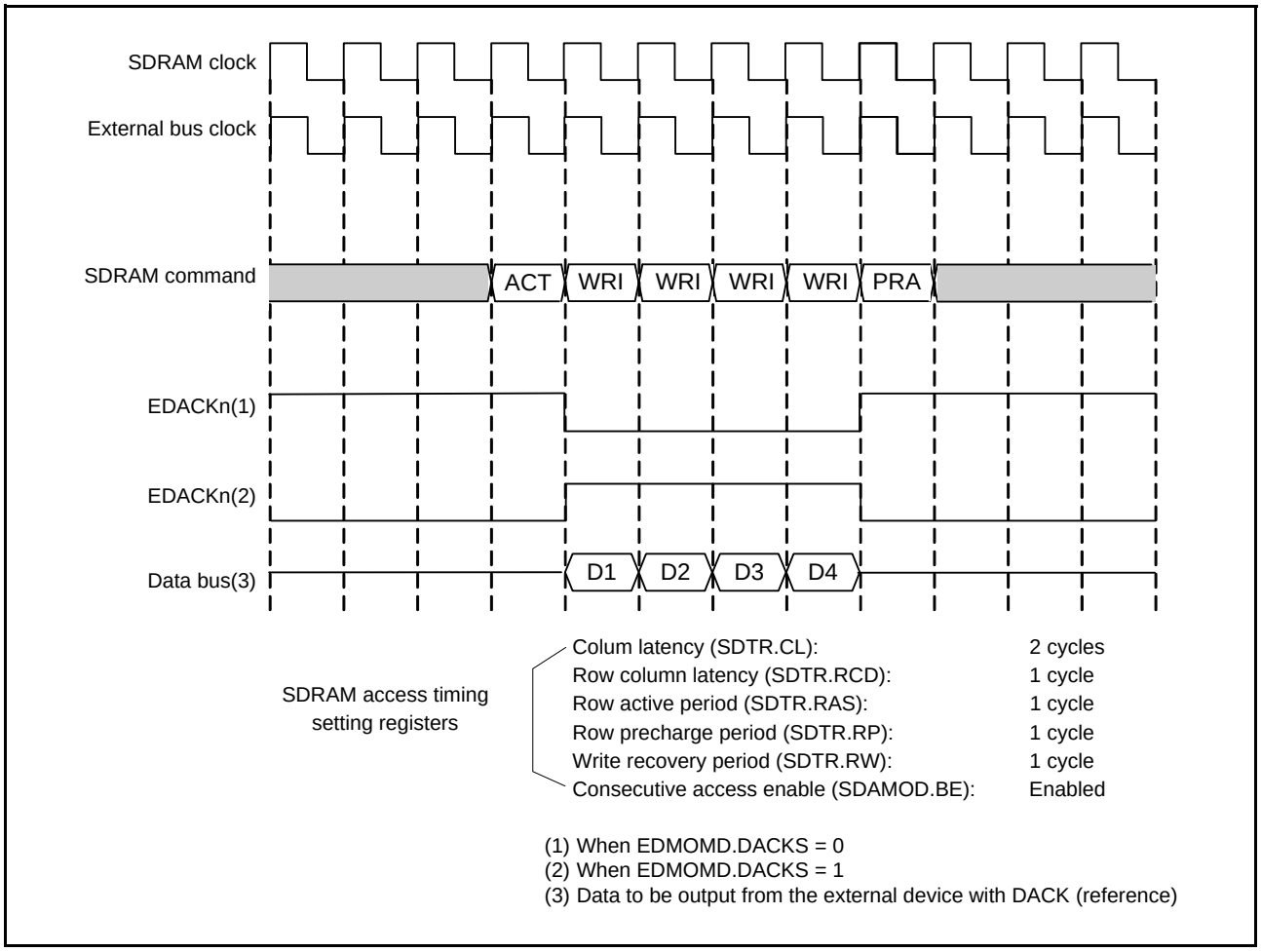


Figure 18.34 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Enabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode

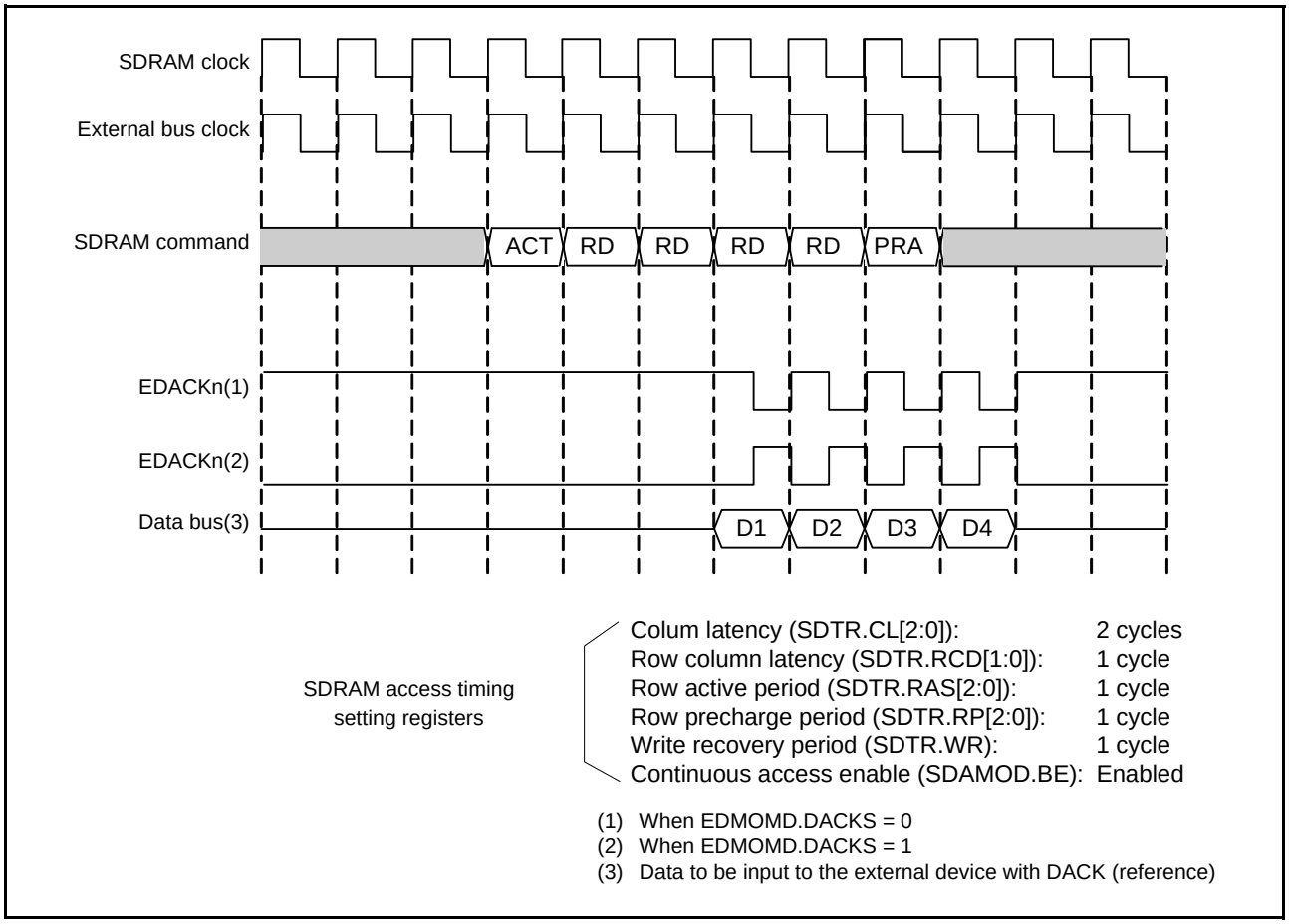


Figure 18.35 Operation Example in Block-Transfer (SDRAM Area Read: Continuous Access Enabled, EXDMACn.EDMOMD.DACKSEL = 1) Single Address Mode

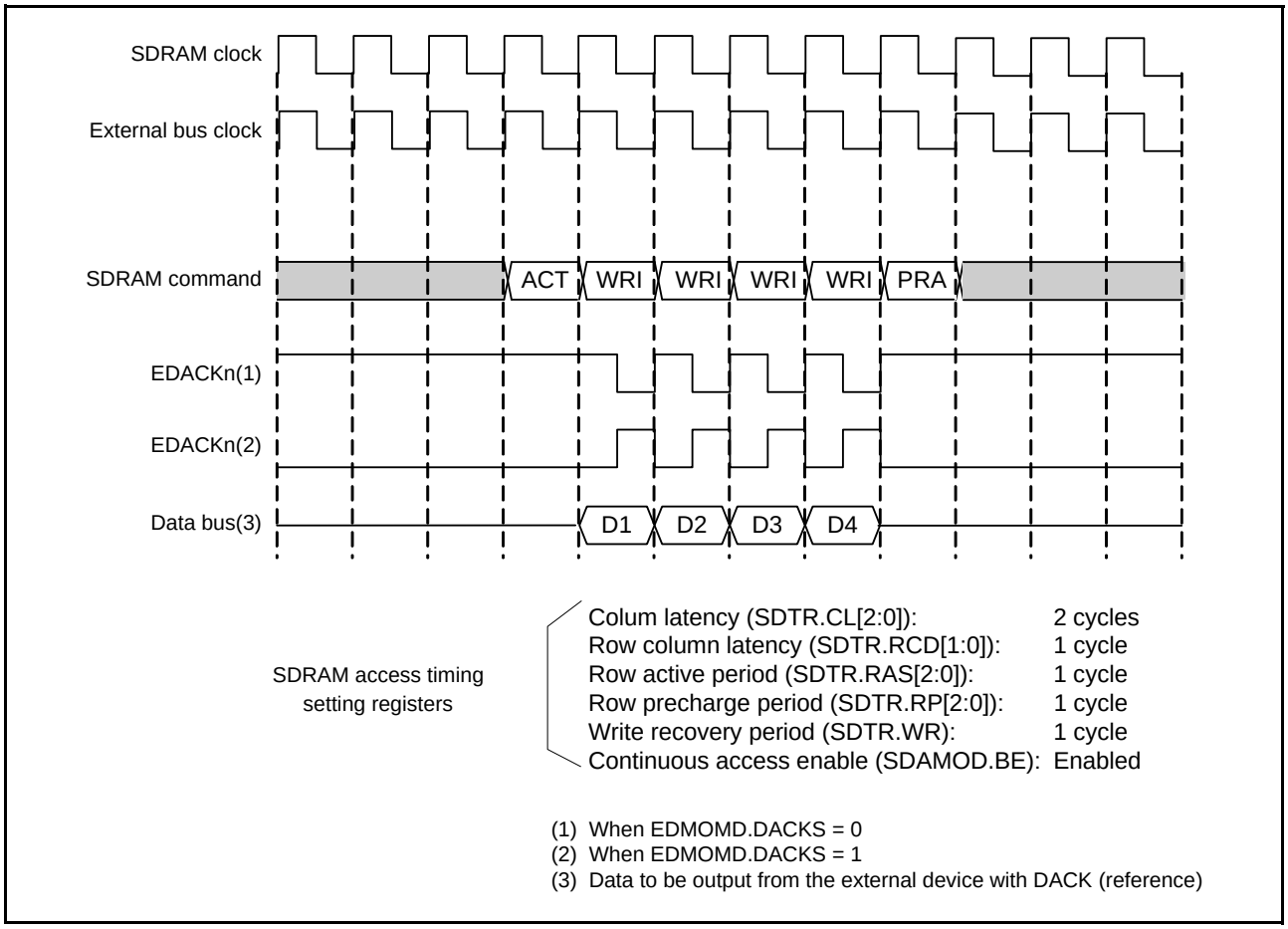


Figure 18.36 Operation Example in Block-Transfer (SDRAM Area Write: Continuous Access Enabled, EXDMACn.EDMOMD.DACKSEL = 1) Single Address Mode

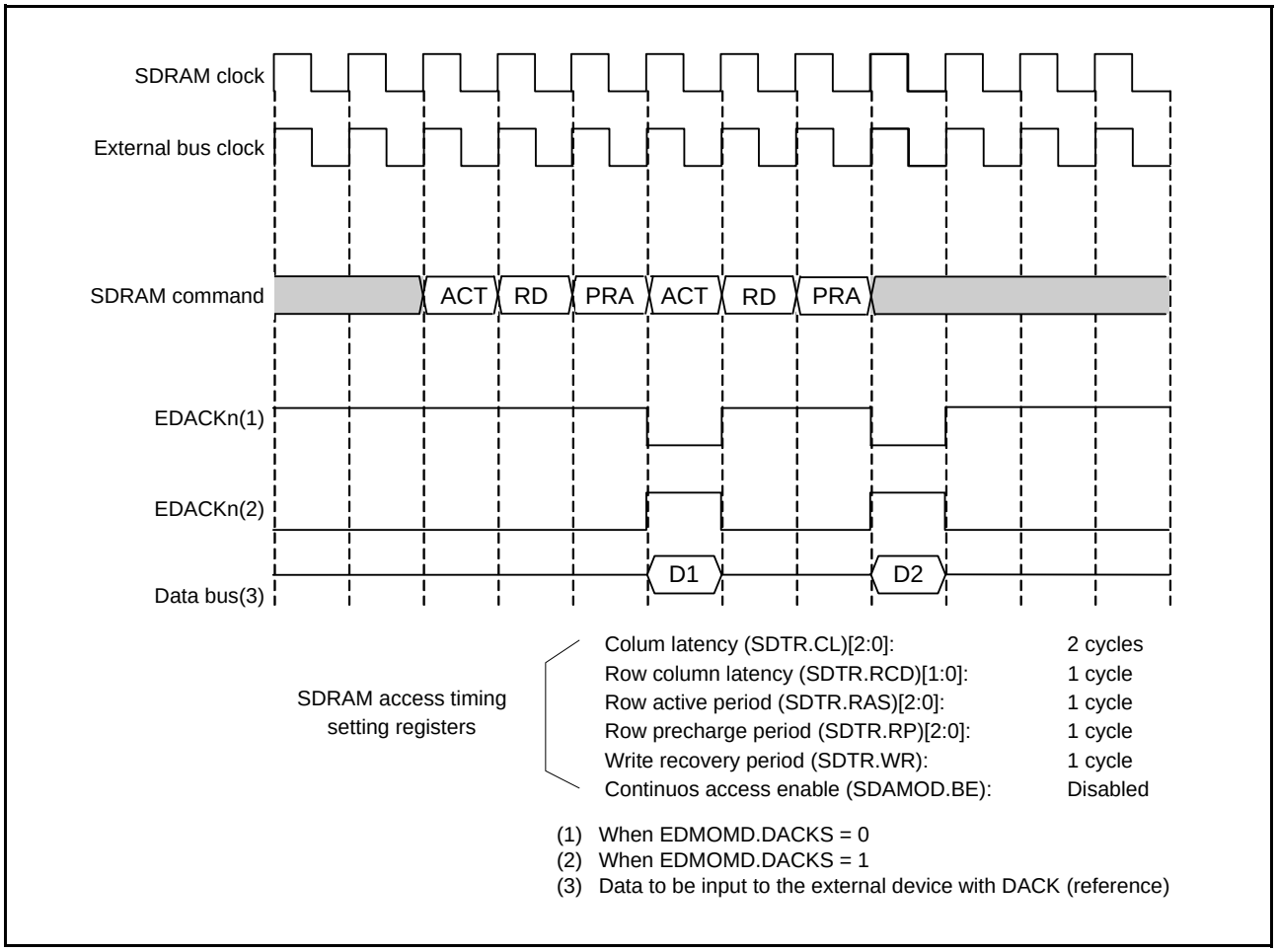


Figure 18.37 Operation Example in Block-Transfer (SDRAM Area Read: Continuous Access Disabled, EXDMACn.EDMOMO.DACKSEL = 0) Single Address Mode

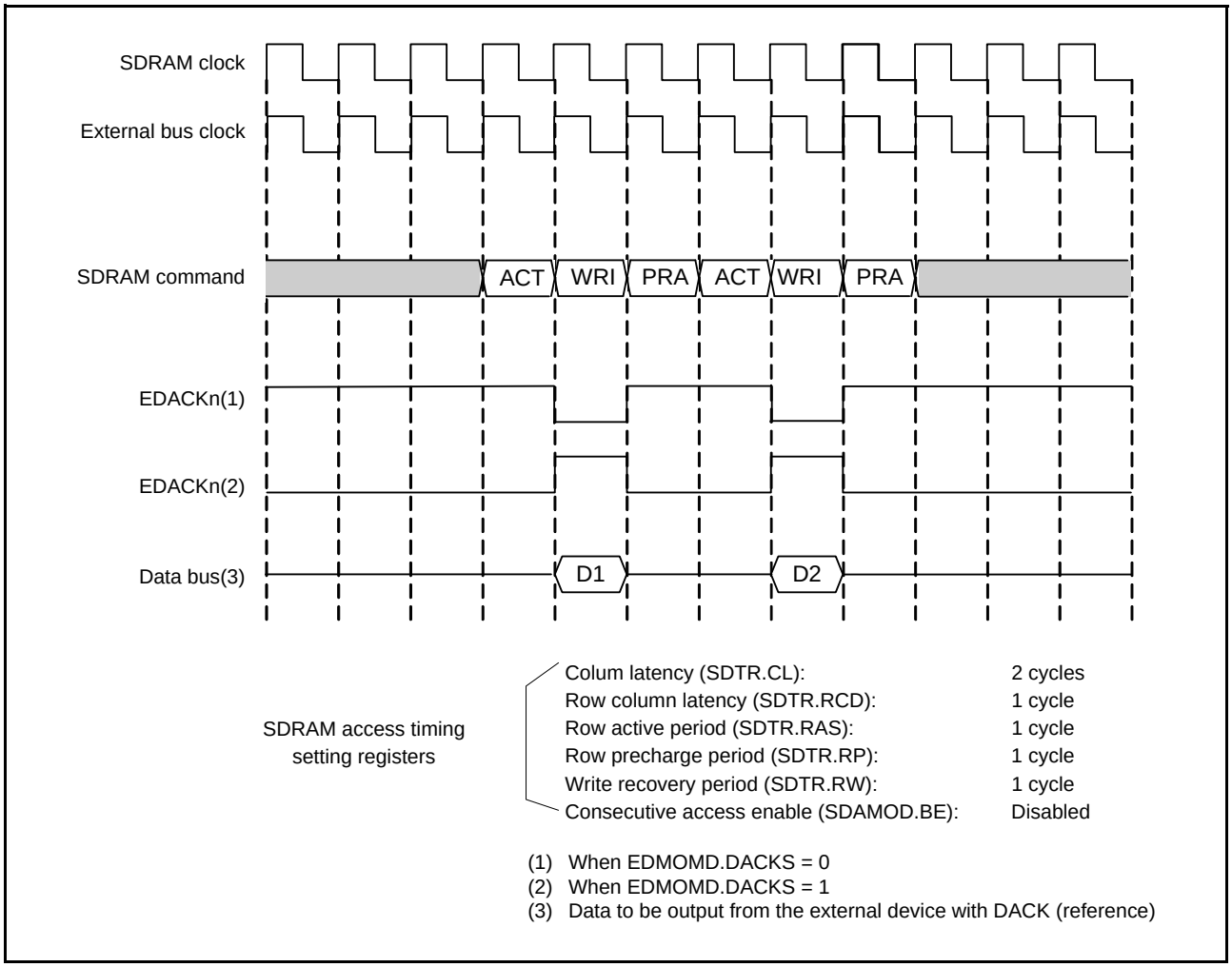


Figure 18.38 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Disabled) Single Address Mode

18.10 Usage Notes

18.10.1 Cluster Buffers

The EXDMAC provides seven 32-bit cluster buffers (CLSBR0 to CLSBR6), in which data is stored in the different manner depending on the transfer size setting (SZ bits in EDMTMD of EXDMACn).
Figure 18.39 shows how data is stored in cluster buffers.

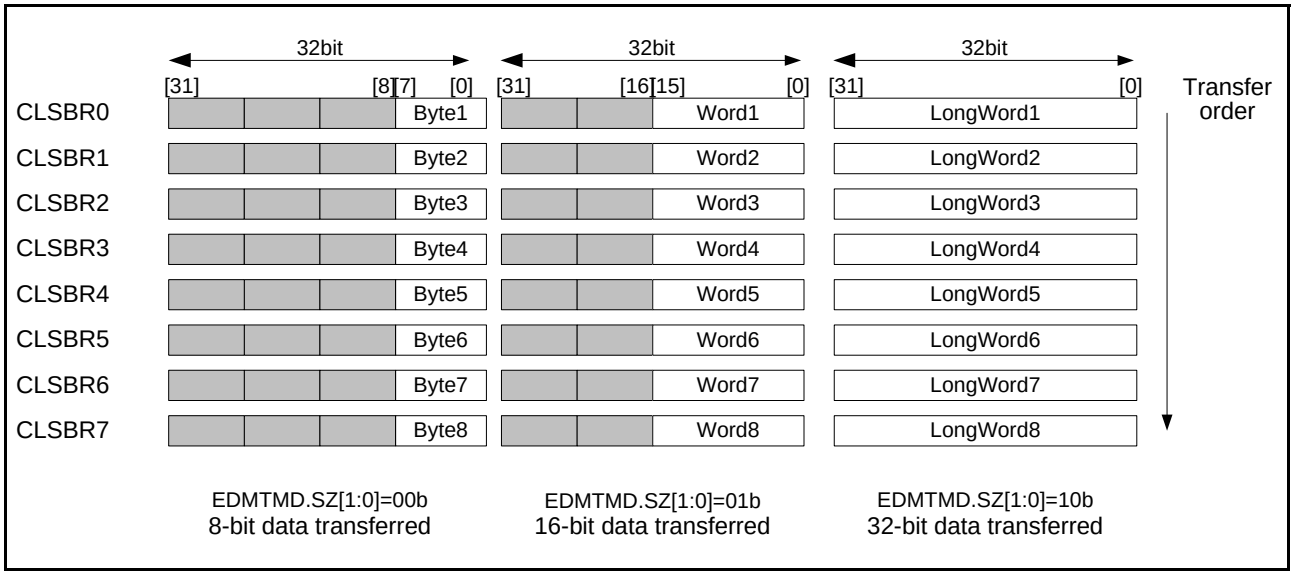


Figure 18.39 Data Storage in Cluster Buffers

18.10.2 Access to the Registers during DMA Transfer

The EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMTMD, EDMOMD, EDMINT, EDMAMD, EDMOFR, and EDMRMD registers of EXDMACn must not be accessed while the ACT bit in EDMSTS of the same channel is set to 1 (DMA operating state) or the DTE bit in EDMCNT of the same channel is set to 1 (DMA transfer enabled).

18.10.3 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.

19. Data Transfer Controller (DTCa)

The RX63N/RX631 Group incorporates a data transfer controller (DTC).
The DTC is activated by an interrupt request to control data transfer.

19.1 Overview

Table 19.1 lists the specifications of the DTC, and Figure 19.1 shows a block diagram of the DTC.

Table 19.1 DTC Specifications

Item	Description
Transfer mode	• Normal transfer mode A single activation leads to a single data transfer. • Repeat transfer mode A single activation 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 repeat size is 256. • Block transfer mode A single activation leads to the transfer of a single block. The maximum block size is 256 data.
Transfer channel	• Channel transfer corresponding to the interrupt source is possible (transferred by DTC activation request from the ICU). • Data of multiple channels can be transferred on a single activation source (chain transfer). • Either “executed when the counter is 0” or “always executed” can be selected for chain transfer.
Transfer space	• In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh excepting reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh excepting reserved areas)
Data transfer units	• Length of a single data: 8, 16, or 32 bits • Number of data for a single block: 1 to 256 data
CPU interrupt source	• An interrupt request can be generated to the CPU on a DTC activation interrupt. • 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.
Read skip	Transfer data read skip can be specified.
Write-back skip	When “fixed” is selected for transfer source address and/or transfer destination address, write-back skip execution is provided.
Lower power consumption function	Module stop state can be specified.

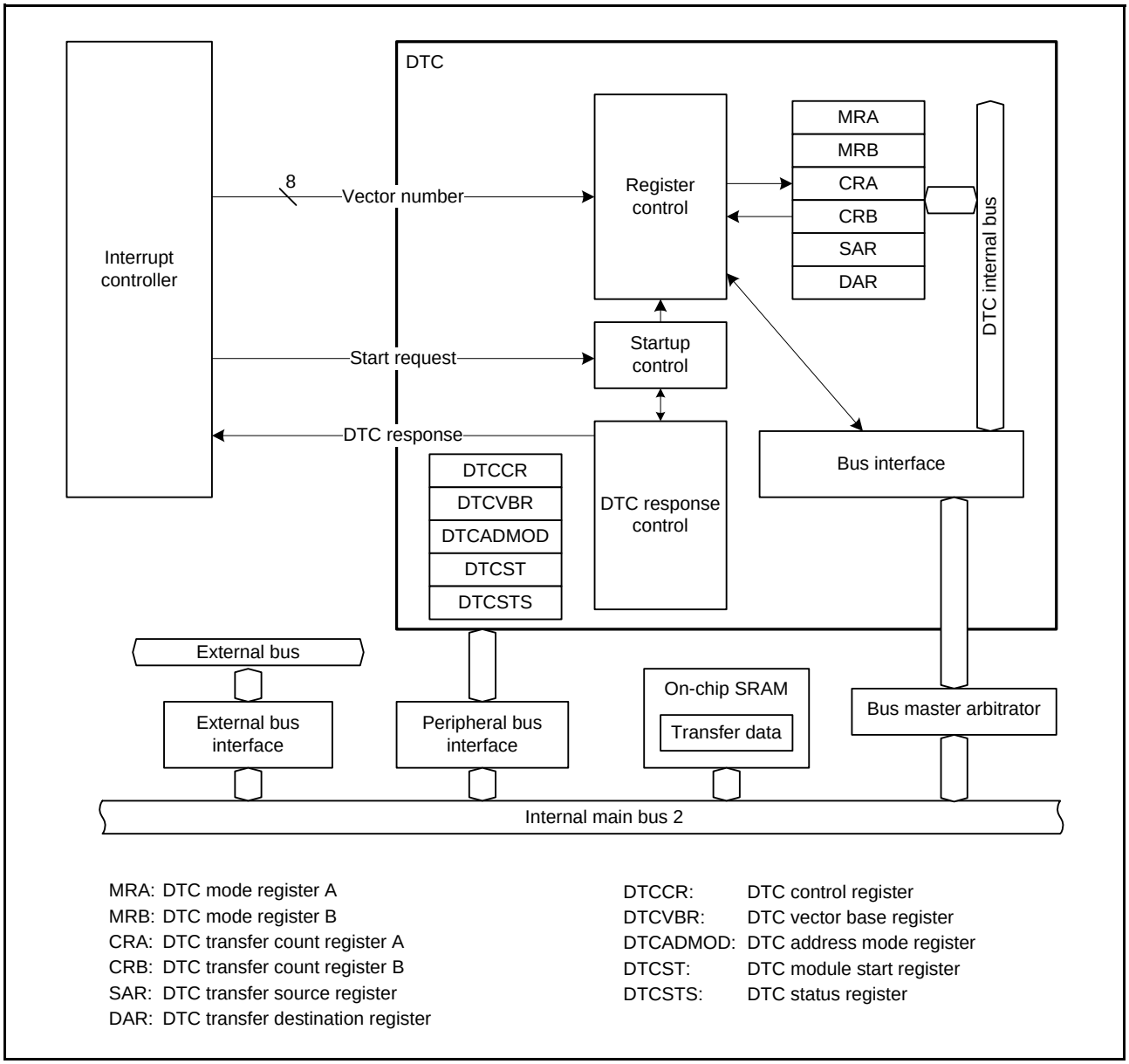


Figure 19.1 Block Diagram of DTC

19.2 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 the DTC internal registers are placed in the RAM area as transfer information data. When an activation request is generated, the DTC reads the transfer information data from the RAM area and set them in the internal registers. After the data transfer ends, the internal register contents are written back to the RAM area as transfer information data.

19.2.1 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	These bits are read as undefined. The write value should be 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: Address in the SAR register is fixed (Write-back to SAR is skipped) 0 1: Address in the SAR register is fixed (Write-back to SAR is skipped) 1 0: SAR value is incremented after data transfer (+1 when SZ[1:0] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: SAR value is decremented after data transfer (–1 when SZ[1:0] bits = 00b, –2 when SZ[1:0] bits = 01b, –4 when SZ[1:0] bits = 10b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: 8-bit (byte) transfer 0 1: 16-bit (word) transfer 1 0: 32-bit (longword) 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 cannot be accessed directly from the CPU.

19.2.2 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	These bits are read as undefined. The write value should be 0.	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: Address in the DAR register is fixed (Write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (Write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer (+1 when SZ[1:0] bits in MRA = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: DAR value is decremented after data transfer (–1 when SZ[1:0] bits in MRA = 00b, –2 when SZ[1:0] bits = 01b, –4 when SZ[1:0] bits = 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 when specified data transfer is completed 1: An interrupt request to the CPU is generated each time DTC data transfer is performed	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed continuously 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 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, see Table 19.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the startup source flag 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, see section 19.4.6, Chain Transfer.

19.2.3 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)

	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

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 cannot be accessed directly from the CPU.

19.2.4 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)

	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

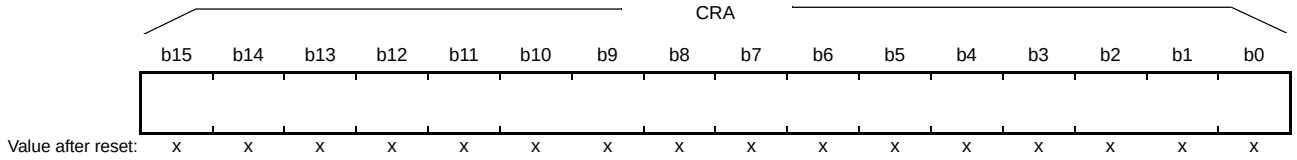
x: Undefined

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 cannot be accessed directly from the CPU.

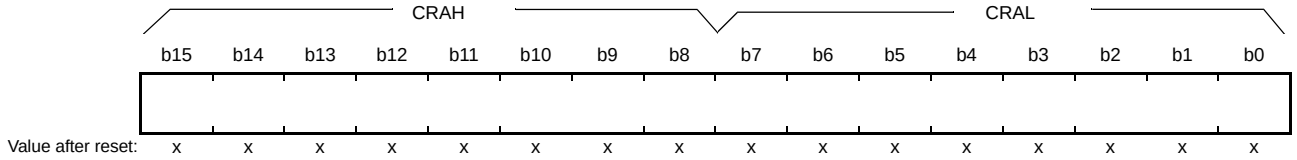
19.2.5 DTC Transfer Count Register A (CRA)

Address(es): (inaccessible directly from the CPU)

- Normal transfer mode



- Repeat transfer mode/block transfer mode



x: Undefined

Note: • The function depends on transfer mode.

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count.	—
CRAH	Transfer Counter A Upper Register		—

Note: • Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

CRA cannot be accessed directly from the CPU.

(1) Normal transfer mode (MD[1:0] bits in MRA = 00b)

CRA 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 (MD[1:0] bits in MRA = 01b)

The CRAH register retains 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 transferred to CRAL.

(3) Block transfer mode (MD[1:0] bits in MRA = 10b)

The CRAH register retains 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 transferred to CRAL.

19.2.6 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 is used to set the block transfer count for block transfer mode.
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.
CRB cannot be accessed directly from the CPU.

19.2.7 DTC Control Register (DTCCR)

Address(es): 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 Data Read Skip Enable	0: Transfer data read is not skipped 1: Transfer data 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

RRS Bit (DTC Transfer Data Read Skip Enable)

The DTC vector number is always compared with the vector number in the previous startup process.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred data. However, when the previous transfer was chain transfer, the transferred data is always read regardless of the value of 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 data is always read regardless of the value of RRS bit.

19.2.8 DTC Vector Base Register (DTCVBR)

Address(es): 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
					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

DTCVBR is used to set the base address for calculating the DTC vector table address.
Write 0 to the lower 12 bits (b11 to b0). These bits are read as 0.
The upper 4 bits (b31 to b28) are ignored, and the address of this register is extended by the value specified by b27.

19.2.9 DTC Address Mode Register (DTCADMOD)

Address(es): 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	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 is used to specify the area accessible by the DTC.

SHORT Bit (Short-Address Mode)

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).

19.2.10 DTC Module Start Register (DTCST)

Address(es): 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 cleared to 0, transfer requests are no longer accepted.

If this bit is cleared to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Before making transition to the module stop state, all-module clock-stop mode, software standby mode, or deep software standby mode, the DTCST bit must be set to 0.

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 19.8, Low-Power Consumption Function, and section 11, Low Power Consumption.

19.2.11 DTC Status Register (DTCSTS)

Address(es): 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-Activating Vector Number Monitoring	These bits indicate the vector number for the activating source when DTC transfer is in progress. The value is only valid if DTC 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 this bit has no effect.	R
b15	ACT	DTC Active Flag	0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

VECN[7:0] Bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, these bits indicate the vector number corresponding to the activating source for the transfer.

When the DTCSTS register is read, the value read from the VECN[7:0] bits is valid if the value of the ACT flag was 1 (indicating DTC transfer in progress) and invalid if the value of the ACT flag was 0 (indicating no current DTC transfer).

For the correspondence between the DTC startup sources and the vector addresses, see Table 15.3, Interrupt Vector Table in section 15, Interrupt Controller (ICUb).

ACT Bit (DTC Active Flag)

This bit indicates the state of DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request

[Clearing condition]

- When transfer by the DTC is completed in response to a transfer request.

19.3 Sources of Activation

The DTC is activated by an interrupt request. Setting the DTCERn.DTCE bit (where n is the interrupt vector number of the given interrupt) of the ICU to 1 selects the corresponding interrupt as an activation source for the DTC.
For the correspondence between the DTC startup sources and the vector addresses, see Table 15.3, Interrupt Vector Table in section 15, Interrupt Controller (ICUb). For startup by software, see section 15.2.5, Software Interrupt Activation Register (SWINTR).
Once the DTC has accepted a startup request, it does not accept another startup request until transfer for that single request is completed, regardless of the priority of the requests. When multiple startup requests are generated during DMAC/DTC transfer, a request with the highest priority on completion of the transfer is accepted. When multiple startup requests are generated while the DTC module start (DTCST) bit in DTCST is 0, a request with the highest priority at the moment when the bit is subsequently set to 1 is accepted.

The following operations are performed for each single round of data transfer (or the last of the consecutive transfers in the case of a chained transfer).

- On completion of a specified round of data transfer, the DTCERn.DTCE bit is cleared 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 startup source is cleared to 0 at the start of data transfer.

19.3.1 Allocating Transfer Data and DTC Vector Table

The DTC reads the start address of the transfer data corresponding to each startup source from the vector table and reads the transfer data starting at that address.
The vector table should be located so that the lower 12 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 data is allocated in the RAM area. In the RAM area, the start address of the transfer data (n) with vector number n should be 4n added to the base address in the vector table.
Transfer data can be allocated in short-address mode (3 longwords) or full-address mode (4 longwords). Use the SHORT bit in DTCADMOD to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 19.2 shows the relationship between the DTC vector table and transfer data.
Figure 19.3 shows the allocation of transfer data in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, see section 19.9.2, Allocating Transfer Data.

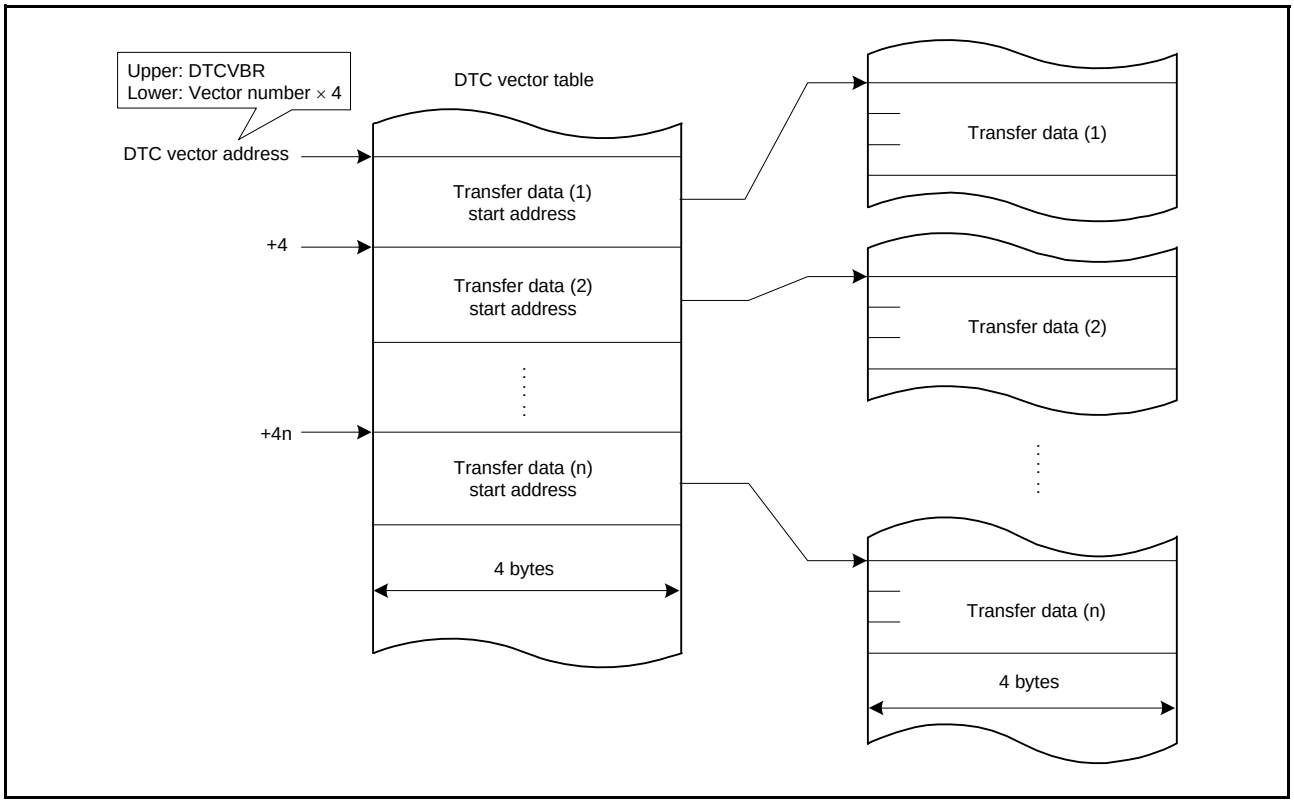


Figure 19.2 DTC Vector Table and Transfer Data

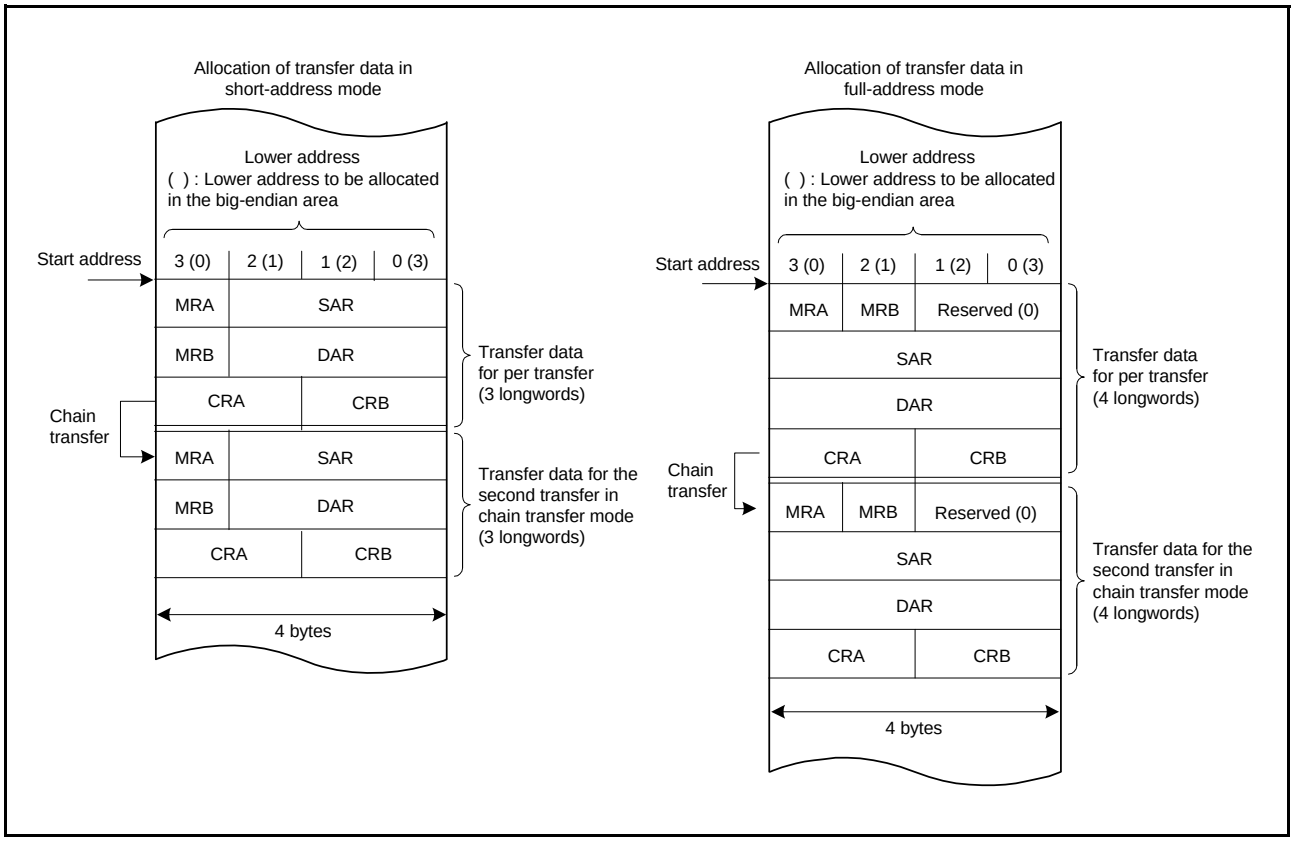


Figure 19.3 Allocation of Transfer Data in the RAM Area

19.4 Operation

The DTC transfers data in accordance with the transfer data. Storage of the transfer data in the RAM area is required before DTC operation.

When the DTC is activated, it reads the DTC vector corresponding to the vector number. Then the DTC reads transfer data from the transfer data store address pointed by the DTC vector, transfers data, and then writes back the transfer data after the data transfer. Storing transfer data 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.

The DTC specifies a transfer source address in SAR and a transfer destination address in DAR. The values of these registers are incremented, decremented, or address-fixed independently after data transfer.

Table 19.2 lists transfer modes of the DTC.

Table 19.2 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on a Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/word/longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/word/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/words/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 CHNE bit in MRB to 1 allows multiple transfers (chain transfer) on a single startup source. Setting the CHNS bit in MRB also enables chain transfer when specified data transfer is completed.

Figure 19.4 shows the operation flowchart of the DTC. Table 19.3 lists chain transfer conditions.

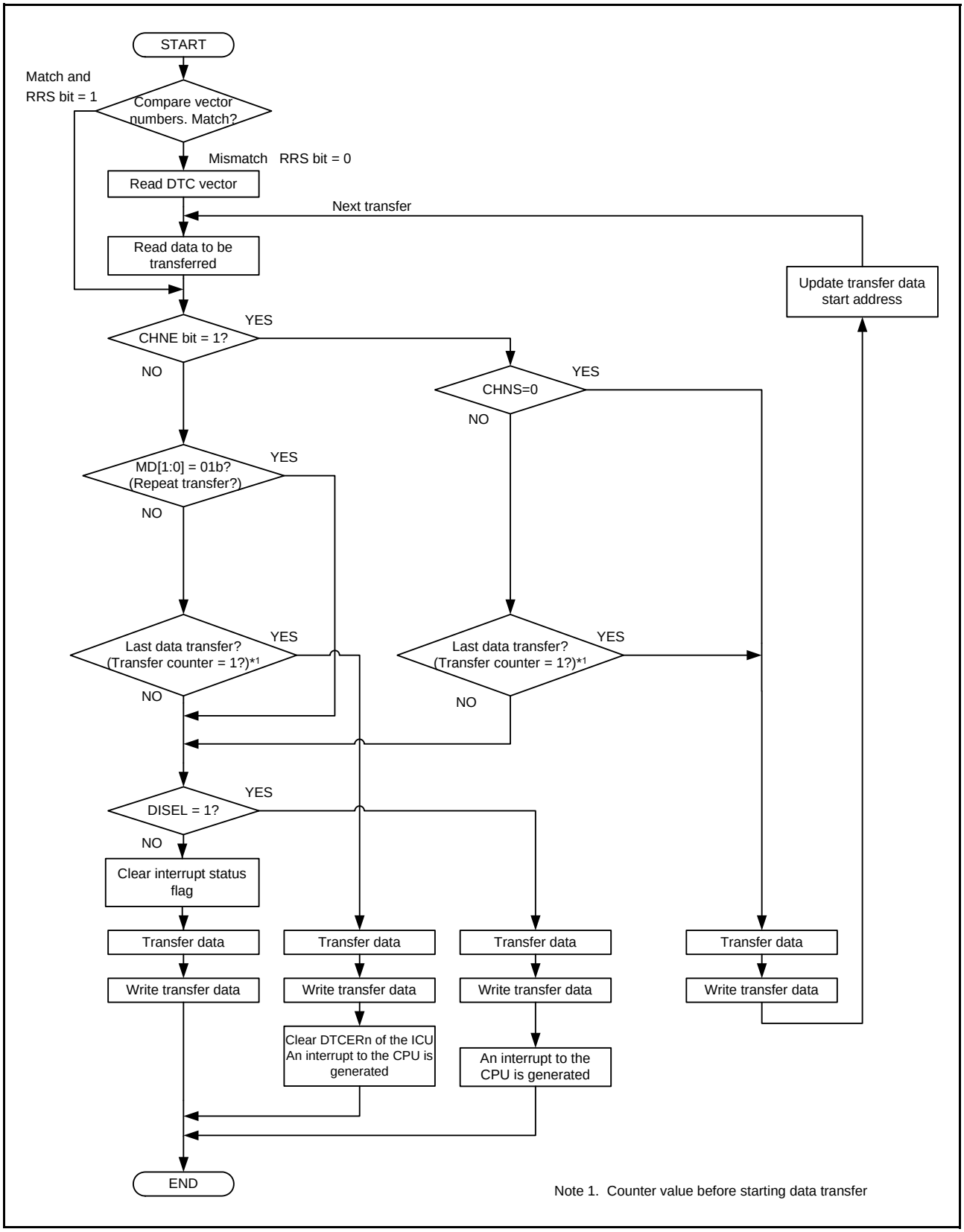


Figure 19.4 Operation Flowchart of the DTC

Table 19.3 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	DTC 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 CHNE bit = 1” is omitted.

19.4.1 Transfer Data Read Skip Function

Vector address read and transfer data read can be skipped by the setting of the RRS bit in DTCCR.
When a DTC startup request is generated, the current DTC vector number is always compared with the DTC vector number in the previous startup process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the vector address and transfer data. However, when the previous transfer was chain transfer, the vector address and transfer data are always 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 data is always read regardless of the value of RRS bit. Figure 19.13 shows an example of transfer data read skip.
To update the vector table and transfer data, set the RRS bit to 0, update the vector table and transfer data, and then set the RRS bit to 1. When the RRS bit is set to 0, the retained vector number is discarded and the vector table and transfer data that are updated in the following startup process are read.

19.4.2 Transfer Data Write-Back Skip Function

When the SM[1:0] bits in MRA or the DM[1:0] bits in MRB are set to “address fixed”, a part of transfer data is not written back. This function is performed independently of the setting of short-address mode or full-address mode. Table 19.4 lists transfer data write-back skip conditions and applicable registers.
The CRA and CRB registers are always written back independently of the setting of short-address mode or full-address mode. Furthermore, in full-address mode, write-back of the MRA and MRB registers are always skipped.

Table 19.4 Transfer Data Write-Back Skip Conditions and Applicable Registers

SM[1:0] Bits in MRA		DM[1:0] Bits in MRB		SAR Register	DAR Register
b3	b2	b3	b2		
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		

19.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source. 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 19.5 lists register functions in normal transfer mode, and Figure 19.5 shows the memory map of normal transfer mode.

Table 19.5 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data
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 in address-fixed mode.

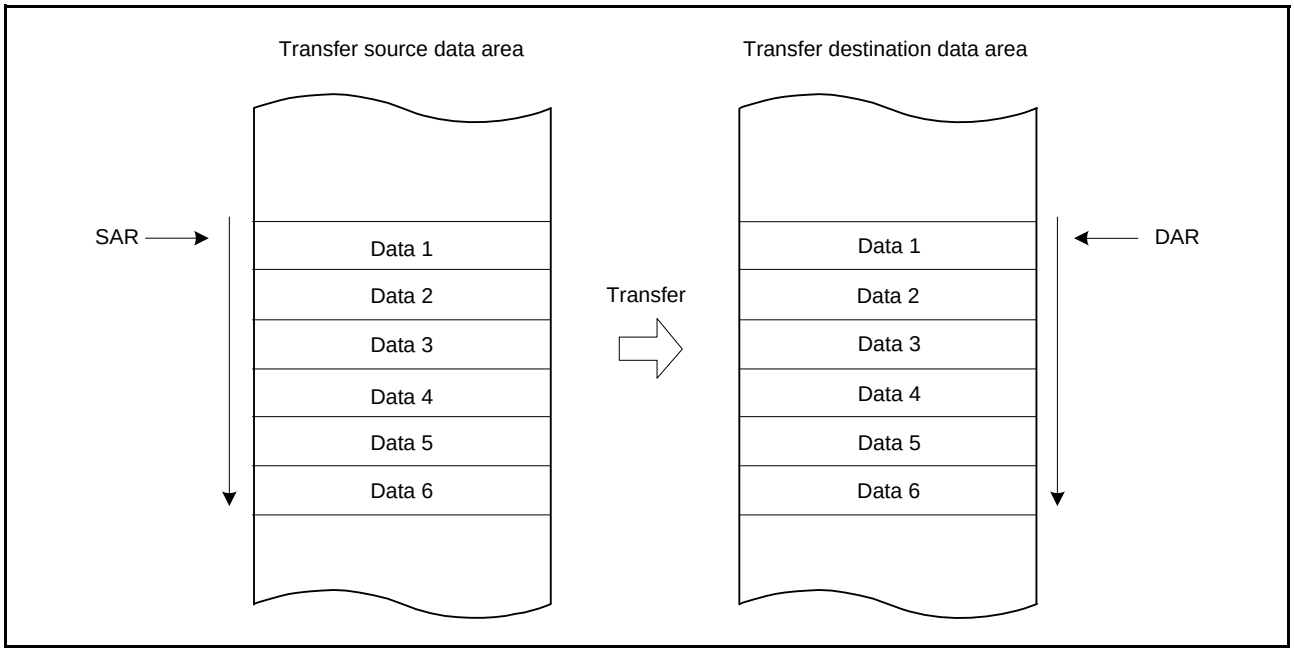


Figure 19.5 Memory Map of Normal Transfer Mode

19.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source.
Specify either transfer source or transfer destination for the repeat area by the DTS bit in MRB. 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 CRAH. Thus the transfer counter does not become 00h, which inhibits generation of interrupt request to the CPU when the DISEL bit in MRB is set to 0 (an interrupt request to the CPU is generated when specified data transfer is completed).
Table 19.6 lists the register functions in repeat transfer mode, and Figure 19.6 shows the memory map of repeat transfer mode.

Table 19.6 Register Functions in Repeat Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data	
		When CRAL is not 1	**When CRAL is 1**
SAR	Transfer source address	Increment/decrement/fixed*1	(When the DTS bit in MRB is 0) Increment/decrement/fixed*1 (When the DTS bit in MRB is 1) SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*1	(When the DTS bit in MRB is 0) DAR register initial value (When the DTS bit in MRB is 1) Increment/decrement/fixed*1
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

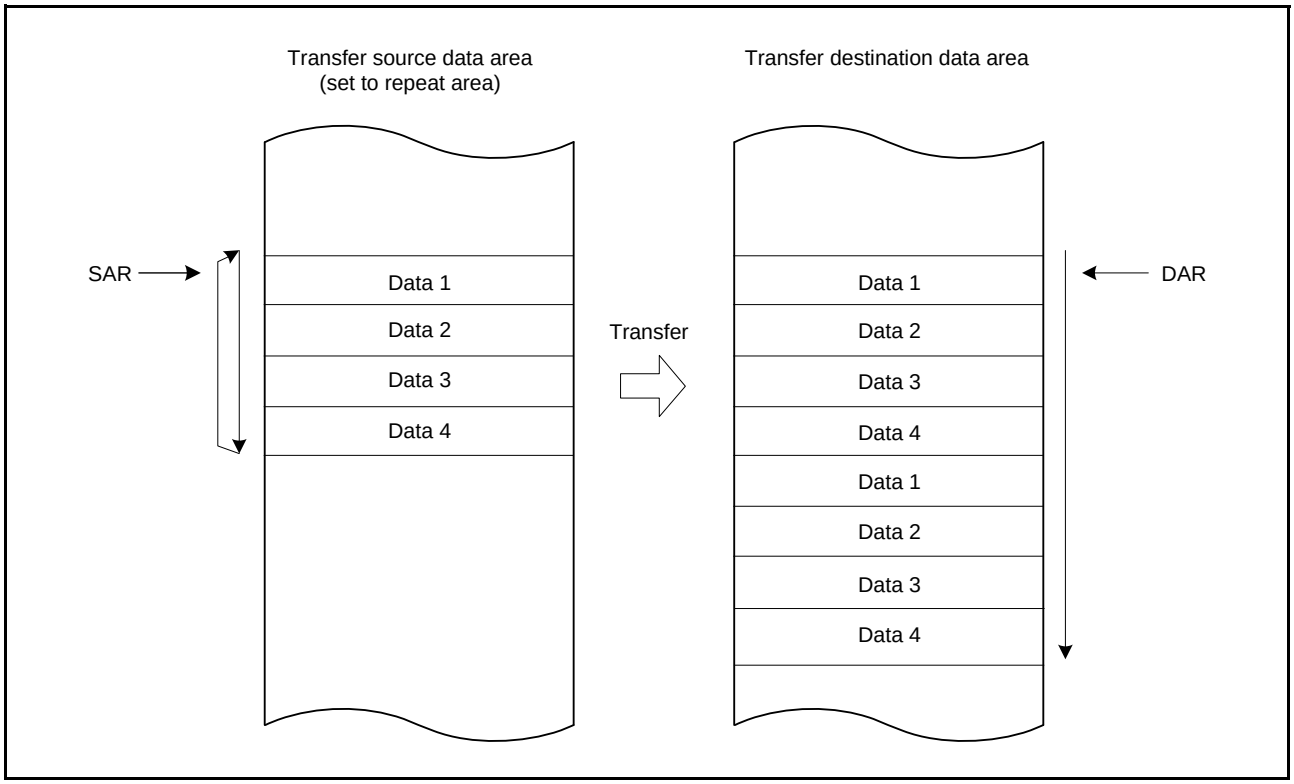


Figure 19.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

19.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single startup source.
Specify either transfer source or transfer destination for the block area by the DTS bit in MRB. The block size can be set to 1 to 256 bytes (or 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 (SAR when the DTS bit = 1 or DAR when the DTS bit = 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 19.7 lists register functions in block transfer mode, and Figure 19.7 shows the memory map of block transfer mode.

Table 19.7 Register Functions in Block Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data
SAR	Transfer source address	(When DTS bit in MRB is 0) Increment/decrement/fixed*1 (When DTS bit in MRB is 1) SAR register initial value
DAR	Transfer destination address	(When DTS bit in MRB is 0) DAR register initial value (When DTS bit in MRB is 1) Increment/decrement/fixed*1
CRAH	Retains block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

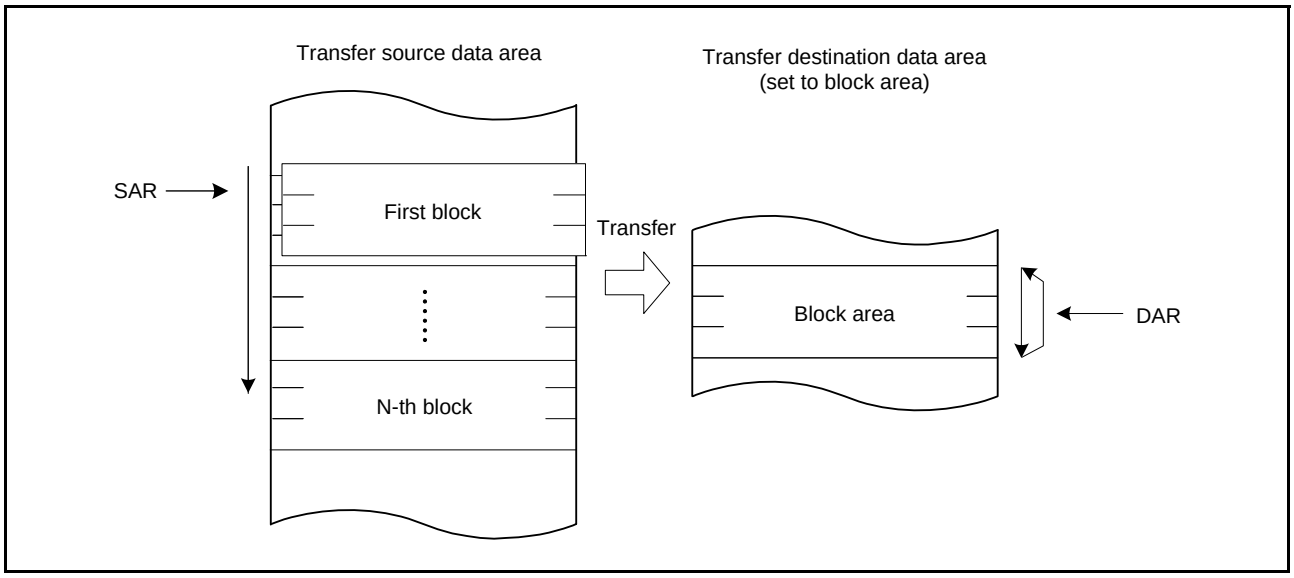


Figure 19.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

19.4.6 Chain Transfer

Setting the CHNE bit in MRB to 1 allows chain transfer to be performed continuously on a single startup source.
If the CHNE and CHNS bits in MRB are set to 1 and 0, respectively, an interrupt request to the CPU is not generated by completion of specified number of rounds of transfer or by setting the DISEL bit in MRB to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed), and data transfer has no effect on the interrupt status flag that has started up the transfer.
The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define data transfer.
Figure 19.8 shows chain transfer operation.

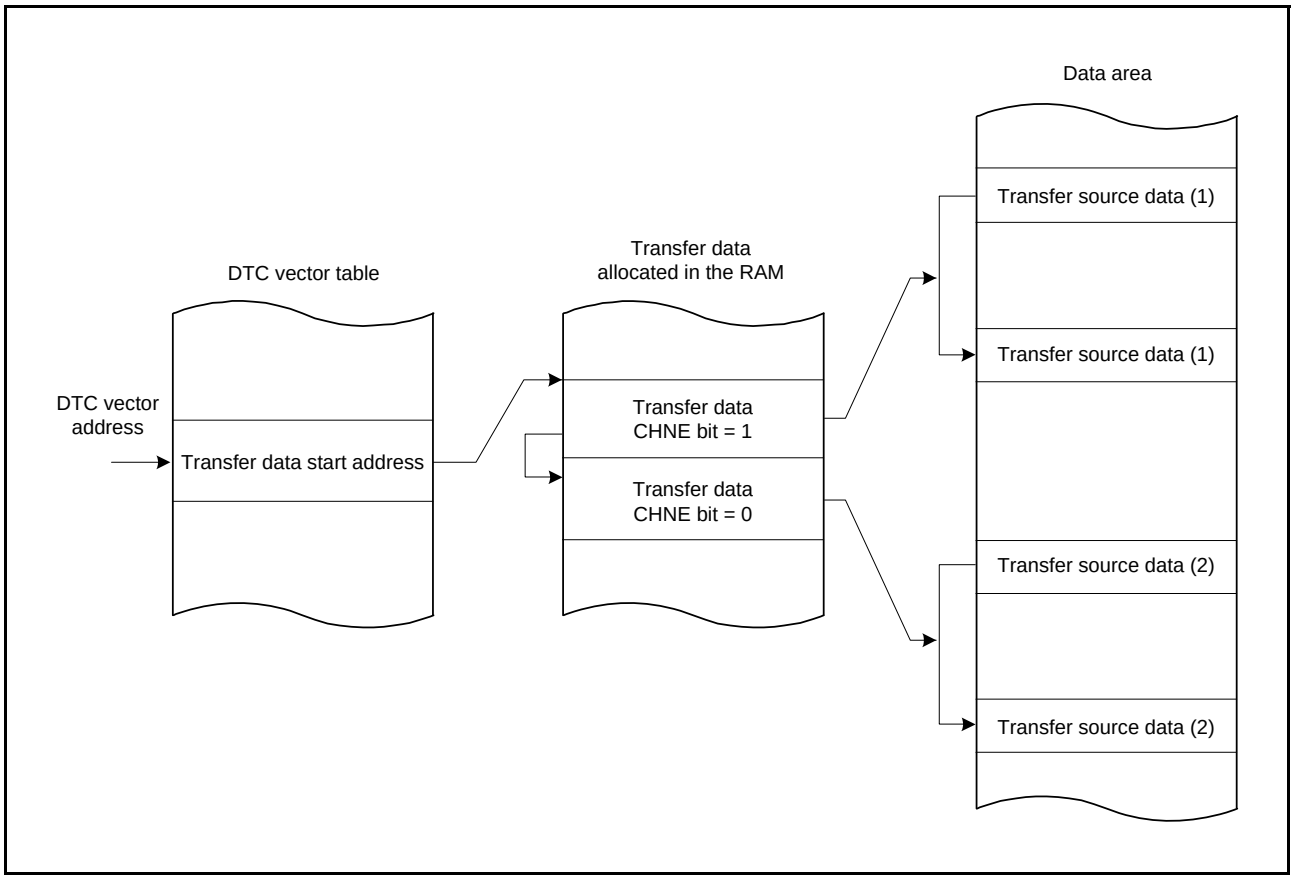


Figure 19.8 Chain Transfer Operation

Writing 1 to the CHNE and CHNS bits in MRB enables chain transfer to be performed only after completion of specified data transfer. In repeat transfer mode, chain transfer is performed after completion of specified data transfer.
For details on chain transfer conditions, see Table 19.3, Chain Transfer Conditions.

19.4.7 Operation Timing

Figure 19.9 to Figure 19.13 show examples of DTC operation timing.

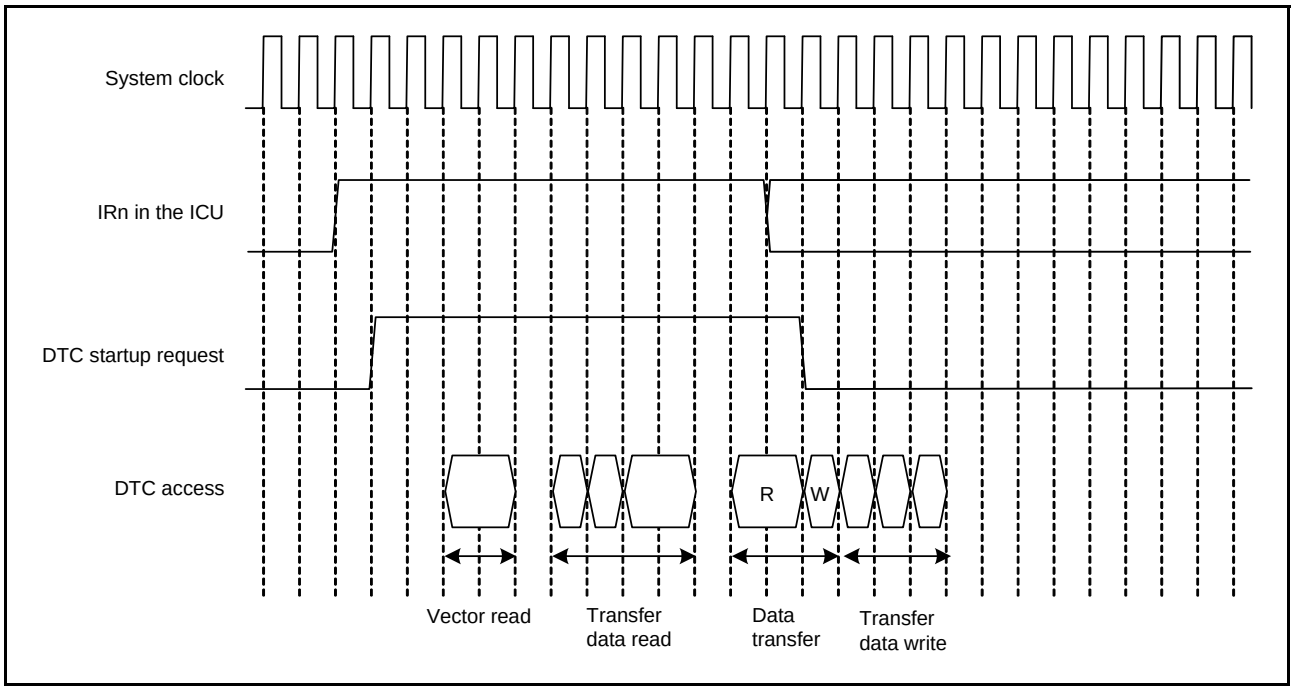


Figure 19.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

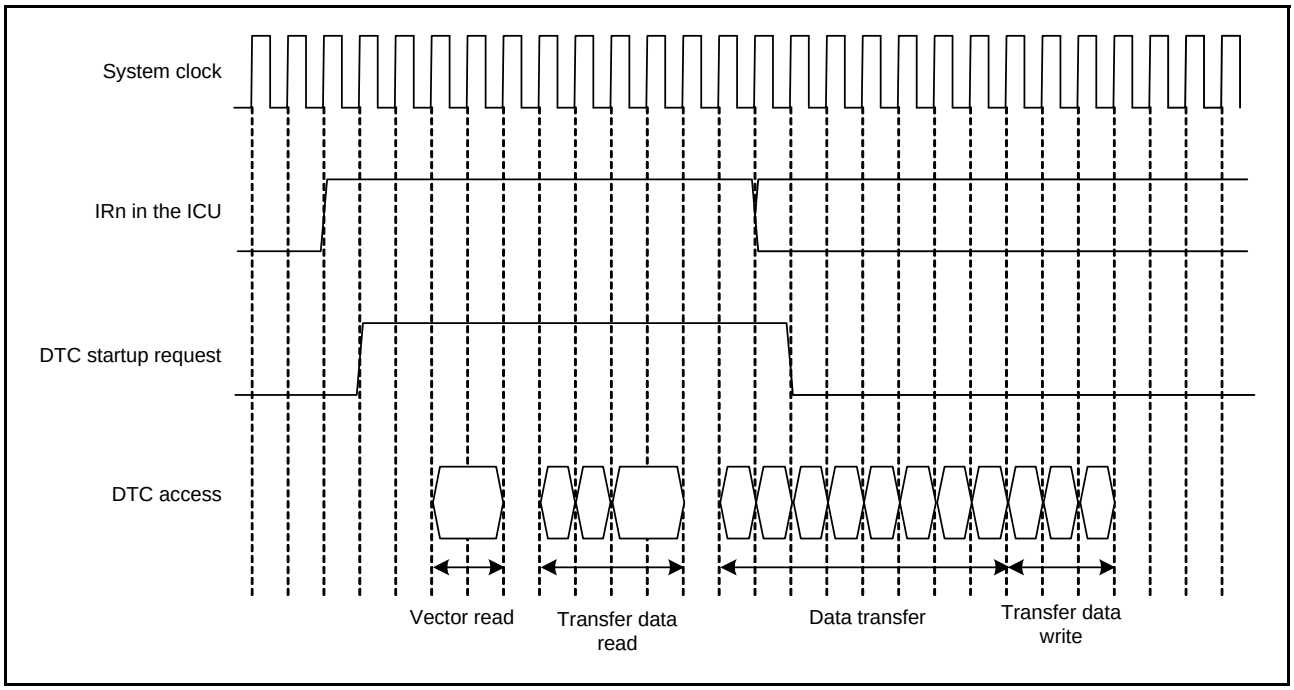


Figure 19.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

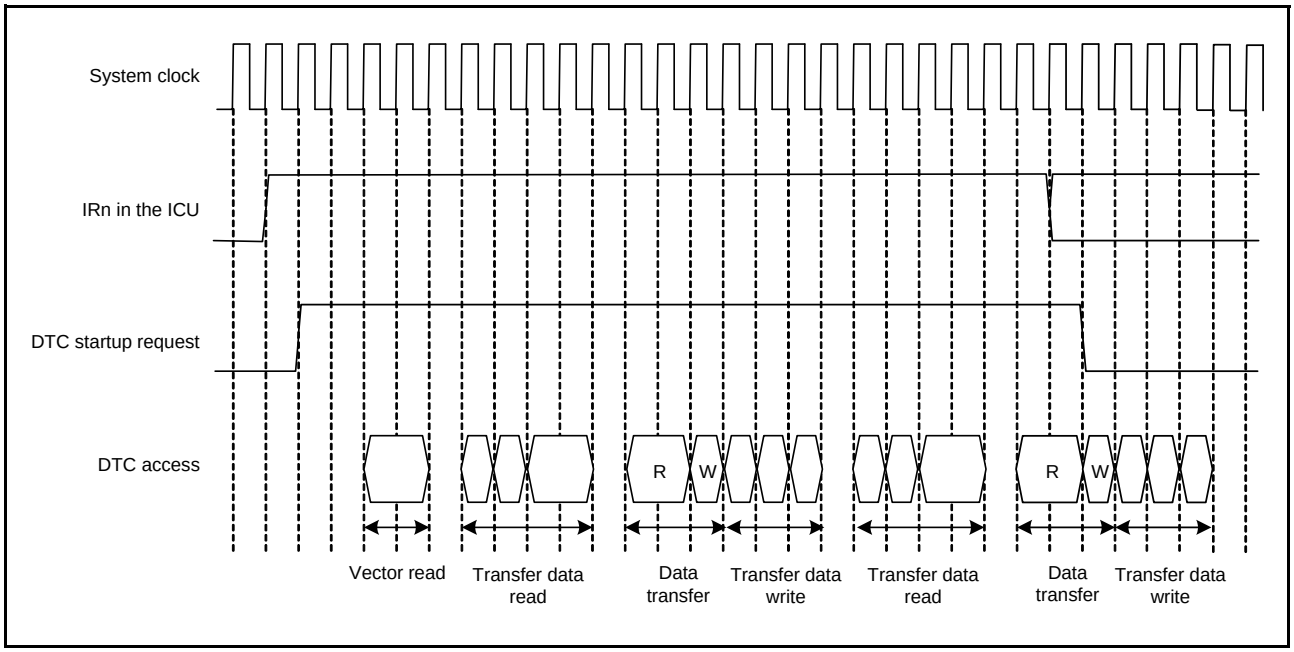


Figure 19.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

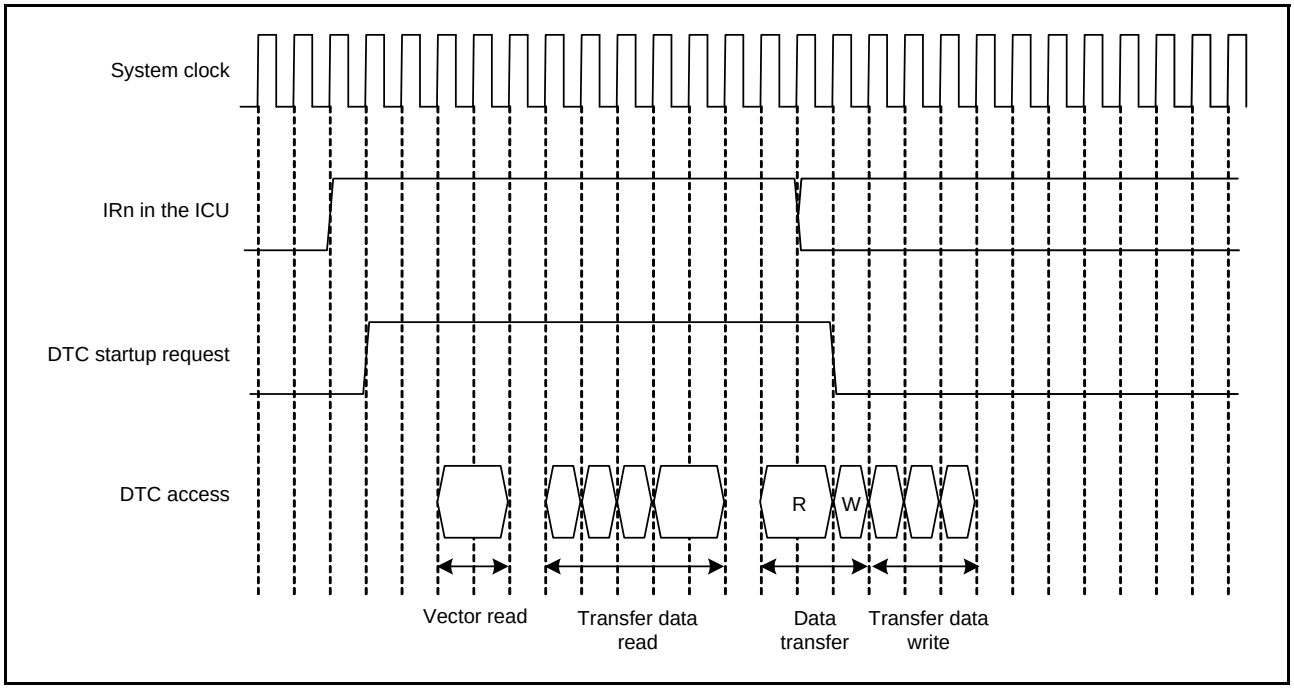
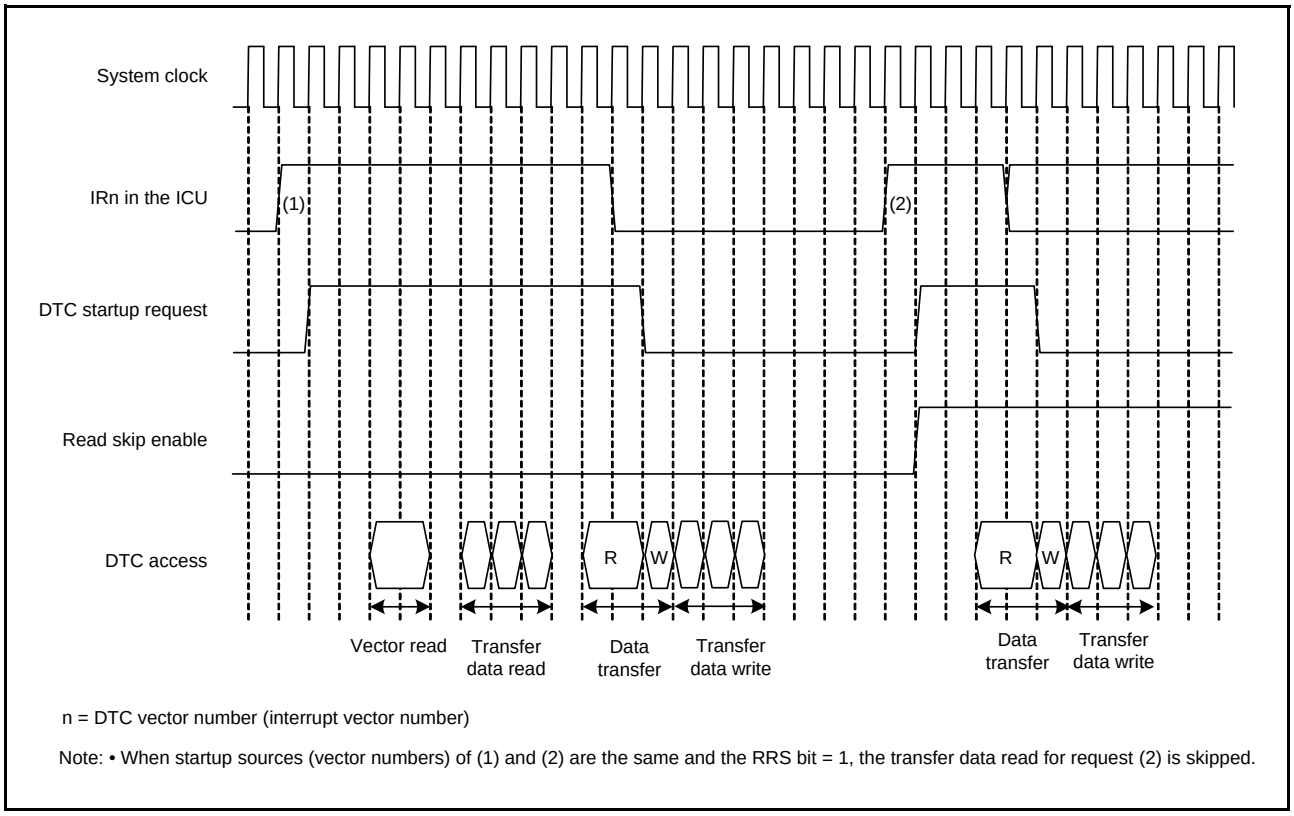


**Figure 19.12 Example (4) of DTC Operation Timing
(Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)**



n = DTC vector number (interrupt vector number)

Note: • When startup sources (vector numbers) of (1) and (2) are the same and the RRS bit = 1, the transfer data read for request (2) is skipped.

Figure 19.13 Example of Operation when Transfer Information Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the On-Chip RAM, and Transfer Source Data on the Peripheral Module)

19.4.8 Execution Cycles of the DTC

Table 19.8 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 19.4.7, Operation Timing.

Table 19.8 Execution Cycles of the DTC

Transfer Mode	Vector Read		Transfer Data Read			Transfer Data Write			Data Transfer Read	Data Transfer Write	Internal Operation	
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 data 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 mode
Note 5. When SAR or DAR is set to address-fixed mode
Note 6. When SAR and DAR are set to address-fixed mode
Note 7. When the block size is 2 or larger. 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 data storage destination
Ci: Cycles for access to transfer data 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 Data 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, see section 44, RAM, section 45, ROM (Flash Memory for Code Storage), section 5, I/O Registers, and section 16.2.5, External Bus.

19.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer data read and transfer data write. While transfer data is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.
For bus arbitration, see section 16, Buses.

19.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).
Figure 19.14 shows the procedure to set the DTC.

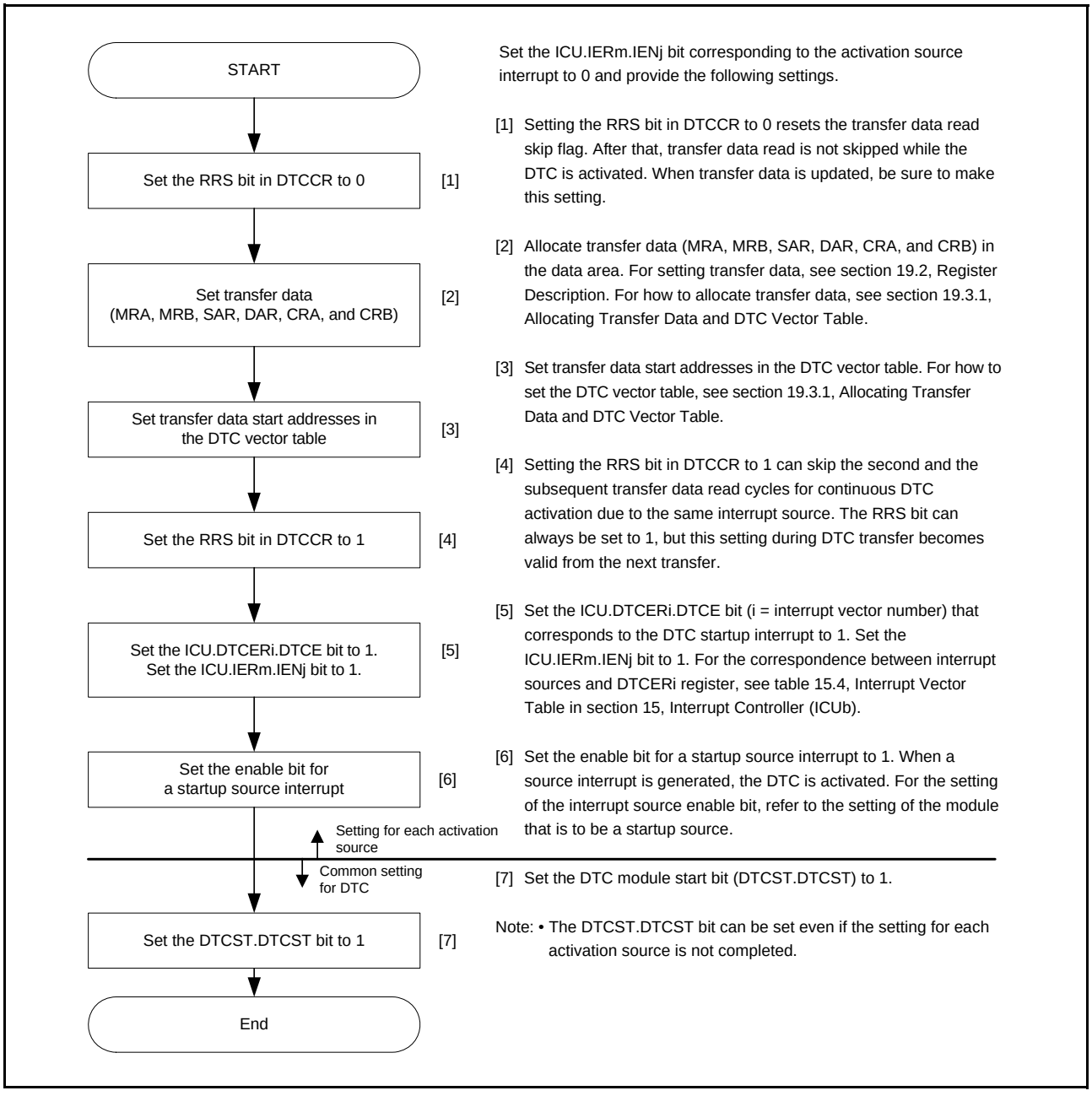


Figure 19.14 Procedure to Set the DTC

19.6 Examples of DTC Usage

19.6.1 Normal Transfer

As an example of DTC usage, its employment in the transfer of 128 bytes of data by an SCI is described below.

(1) Transfer Data Set

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE bit = 0 and MRB.DISEL bit = 0). The MRB.DTS bit can be set to any value. Set the address of the RDR register in the SCIm (m = 0 to 12) 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

The address where the transfer-control information for use with the RXI starts is set in the vector table for the DTC.

(3) ICU Set and DTC Module Activation

Set the corresponding ICU.DTCERi.DTCE bit to 1 and the ICU.IERi.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(4) SCI Set

Enable the receive end interrupt (RXI) by setting the SCR.RIE bit in the SCIm 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 activate the DTC. The DTC transfers the received byte from the RDR of the SCIm to RAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt Handling

After 128 rounds of data transfer have been completed and the value in the CRA register becomes 0, an RXI interrupt request is generated for the CPU. Processing for completion is performed in the processing routine for this interrupt.

19.6.2 Chain Transfer

As an example of chained transfer by the DTC, its employment in the output of pulses by a PPG is described below. Chained 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 chained transfer, repeat transfer mode for transfer to the PPGm.NDRH and PPGm.NDRL registers (m = 0, 1) is specified. For the second half, normal transfer mode for transfer to the MTUm.TGR registers (m = 0 to 5) is specified. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of rounds of transfer are restricted to the second half of the chained transfer (transfer while MRB.CHNE = 0).
An example of how to use the compare-match interrupt for an MTUm.TGRA register (for m = 0 to 4) as an activating source for the DTC is provided below.

(1) First Transfer Data Set

Settings are made for transfer to the PPGm.NDRH and PPGm.NDRL registers. In the MRA register, make the settings to select incrementation of the source address (MRA.SM[1:0] = 10b), transfer in repeated-transfer mode (MRA.MD[1:0] = 01b), and word-sized transfer (MRA.SZ[1:0] = 01b). In the MRB register, make the settings for the destination address fixed (MRB.DM[1:0] = 00b) and for chained transfer (MRB.CHNE = 1 and MRB.CHNS = 0). Set the source side on the repeat area (MRB.DTS = 1). Set the SAR to the first address of the data table, the DAR register to the address of the PPGm.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 Data Set

Settings are made for transfer to the MTUm.TGRA register. In the MRA register, make the settings to select incrementation of the source address (MRA.SM[1:0] = 10b), transfer in normal mode (MRA.MD[1:0] = 00b), and word-sized transfer (MRA.SZ[1:0] = 01b). In the MRB register, make the settings for the destination address fixed (MRB.DM[1:0] = 00b) and for the single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the SAR register to the first address of the data table, the DAR register to the address of the MTUm.TGRA register, and the CRA register to the size of the data table. The CRB register can be set to any value.

(3) Transfer Data Assignment

Place the transfer-control information for use in transfer to the MTU immediately after the transfer-control information for use in transfer to the PPGm.NDRH and PPGm.NDRL registers.

(4) DTC Vector Table

In the DTC vector table, set the address where the transfer-control information for use in transfer to the PPGm.NDRH and PPGm.NDRL registers starts.

(5) ICU Set and DTC Module Activation

Set the ICU.DTCERi.DTCE bit corresponding to the TGIA interrupt and the ICU.IERi.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(6) MTU Set

In the given MTUm, 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 TGImA interrupt requests.

(7) PPG Set

Set the default output values in the PPGm.PODRH and PPGm.PODRL registers and the next output values in the PPGm.NDRH and PPGm.NDRL registers. In the DDR of the corresponding port m (where m = 1 to 3, 7, 8, A to D, E), set the appropriate bit to 1 so that output from the PPGm.NDRH and PPGm.NDRL registers proceeds. Also, select a compare-match signal of the MTU as the output trigger in the PPGm.PCR (m = 0, 1).

(8) MTU Activation

Set the MTU.TSTR.CSTj (j = 0 to 4) bits to 1 to start counting operation of the MTUm.TCNT counter.

(9) DTC Transfer

Every time a compare-match with the MTUm.TGRA register is generated, next output values are transferred to the PPGm.NDRH and PPGm.NDRL registers and the setting for the next output-trigger period is transferred to the MTUm.TGRA register.

(10) Interrupt Handling

After the specified number of rounds of data transfer has been completed (i.e. when the value in the CRA register of the MTU has become 0), a TGIAm interrupt request is issued for the CPU. Processing for completion is performed in the processing routine for this interrupt.

19.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the counter = 0. Repeat transfer of a transfer count of 256 or more is enabled by the re-setting for the first data transfer.
The following shows an example of configuring a 128-Kbyte input buffer, where the input buffer is set so that its lower address starts with 0000h. Figure 19.15 shows a chain transfer when the counter = 0.

1. Set normal transfer mode for input data for the first data transfer. Set the following:
   Transfer source address: Fixed, CRA = 0000h (65,536 times), CHNE bit = 1 (chain transfer enabled) in MRB, CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0) in MRB, and DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed) in MRB.
2. Prepare the upper 8-bit address 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). For example, when setting the input buffer to 200000h to 21FFFFh, prepare 21h and 20h.
3. For the second data transfer, set repeat transfer mode (source side: repeat area) for re-setting the transfer destination address of the first data transfer. Specify the upper 8 bits of DAR in the first transfer data area for the transfer destination. At this time, set CHNE bit = 0 (chain transfer disabled) in MRB and DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed) in MRB. When setting the input buffer mentioned above to 200000h to 21FFFFh, set the transfer counter to 2.
4. The first data transfer is performed by an interrupt 65,536 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 21h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
5. In succession, the first data transfer is performed by an interrupt 65,536 times specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 20h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
6. Steps 4 and 5 above are repeated infinitely. Since the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

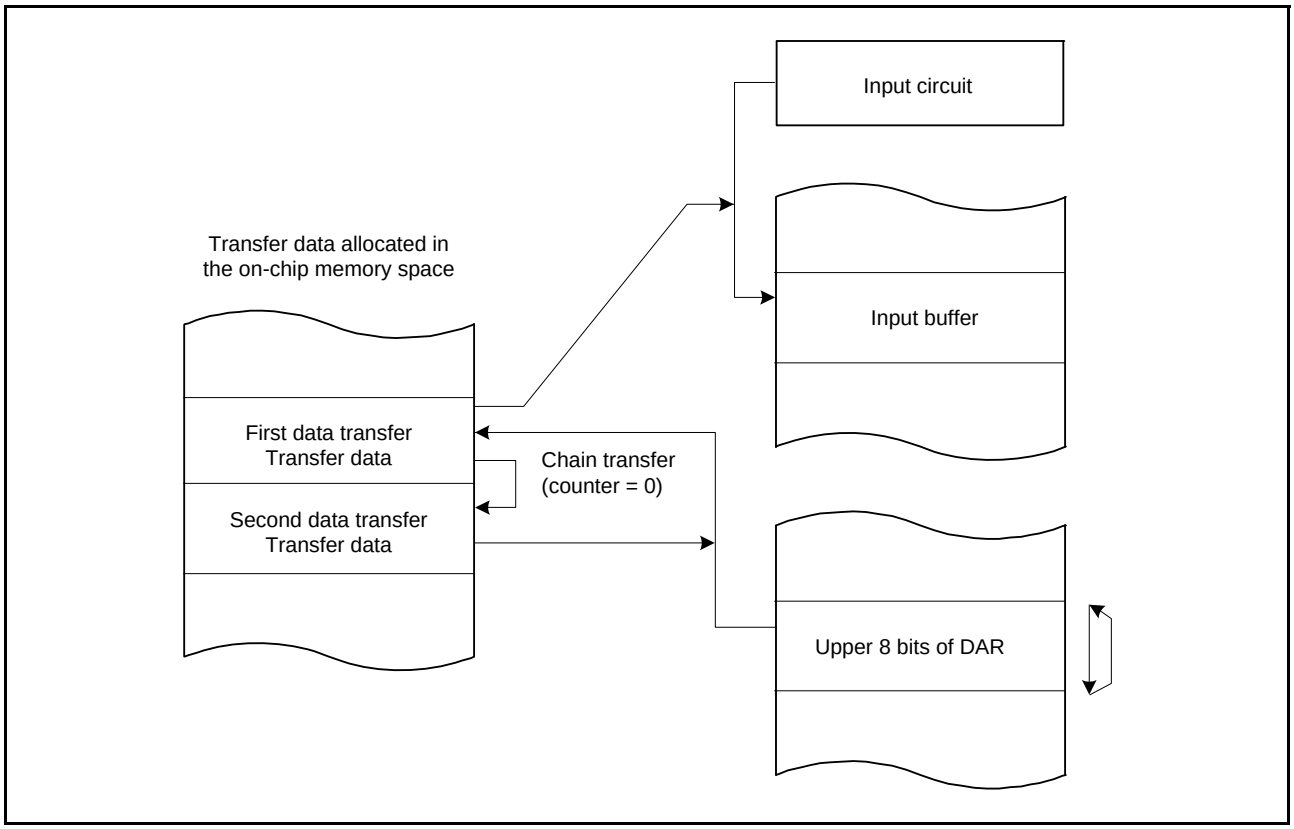


Figure 19.15 Chain Transfer when Counter = 0

19.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the DISEL bit in MRB set to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC startup source. Such interrupts to the CPU are controlled according to the PSWI bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt Controller.

19.8 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 DTCST bit in DTCST to 0 (the DTC 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 DTC. If DTC transfer is in progress at the time a 1 is written to the MSTPA28 bit, the transition to the module stop state proceeds after DTC transfer has ended. While the MSTPA28 bit is 1, accessing the DTC registers are prohibited. Writing 0 to the MSTPA28 bit releases the DTC 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 DTC 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 DTC 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 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 DTC 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 DTC transfer.

(4) Notes on Low-Power Consumption Function

For the WAIT instruction and the register setting procedure, section 11.7.6, Timing of Wait Instructions in section 11, Low Power Consumption.
To perform DTC transfer after returning from low-power consumption mode, set the DTCST bit in DTCST 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 DTC startup request, specify the CPU as the interrupt request destination in accordance with the description in section 15.5.3, Selecting Interrupt Request Destinations in section 15, Interrupt Controller (ICUb), and then execute the WAIT instruction.

19.9 Usage Notes

19.9.1 Transfer Data Start Address/Source Address/Destination Address

Be sure to set multiples of 4 for the transfer data start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

19.9.2 Allocating Transfer Data

Allocate transfer data in the memory area according to the endian of the area as shown in Figure 19.16.
For example, when writing CRA and CRB setting data with 16 bits in big endian, write the CRA setting data to lower address 0 and the CRB setting data to lower address 2. In little endian, write the CRB setting data to lower address 0 and the CRA setting data to lower address 2. When writing CRA and CRB setting data with 32 bits, place the CRA setting data at the MSB side and the CRB setting data at the LSB side regardless of endian, and then write the data to lower address 0.

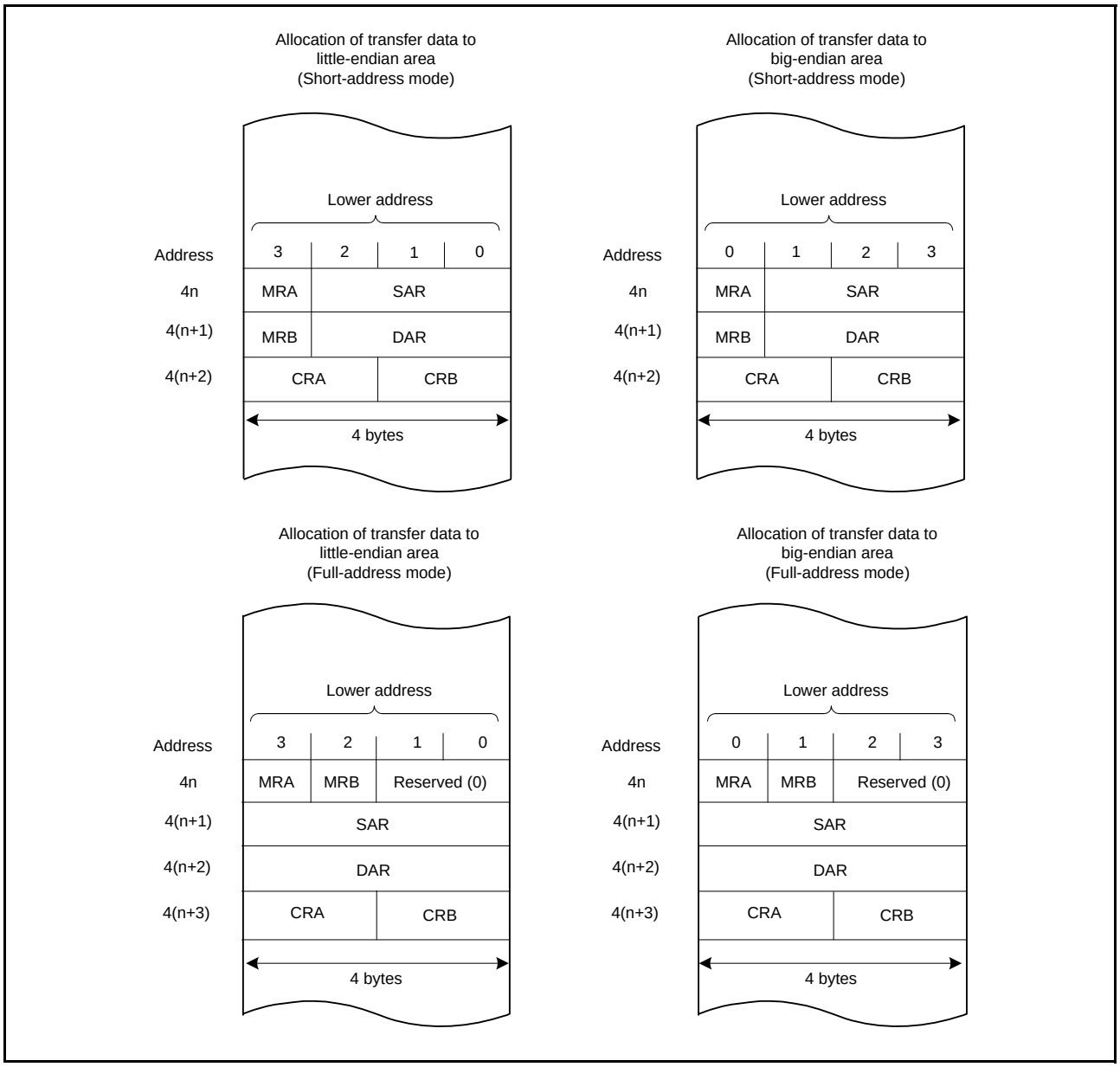


Figure 19.16 Allocation of Transfer Data

19.9.3 Setting the DTC Activation Enable Register (ICU.DTCERn) of the Interrupt Controller

The DTC activation enable register (DTCERn in the ICU) must be set while the DTCST bit in DTCST is 0 (DTC module stop). Moreover, the DMAC should not be activated by setting the DMAC activation request select register (ICU.DMRSRn (n = number of DMAC channel)) to the same vector number that has been specified by setting the ICU.DTCERn register 1 (DTC transfer enable). For details on the ICU.DTCERn and ICU.DMRSRn registers (n = number of DMACA channel), refer to section 15, Interrupt Controller (ICUb).

20. I/O Ports

20.1 Overview

The I/O ports of the RX63N/RX631 Group function as a programmable I/O port, an I/O pin of a peripheral module, an input pin for an interrupt, or a bus control pin.
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 21, Multi-Function Pin Controller (MPC).
The configuration of the I/O ports differs depending on the package. Table 20.1 shows the specifications of I/O ports, and Table 20.2 list the port functions.

Table 20.1 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 P57	8	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 P87	8	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	134	Total of Pins	112	Total of Pins	79

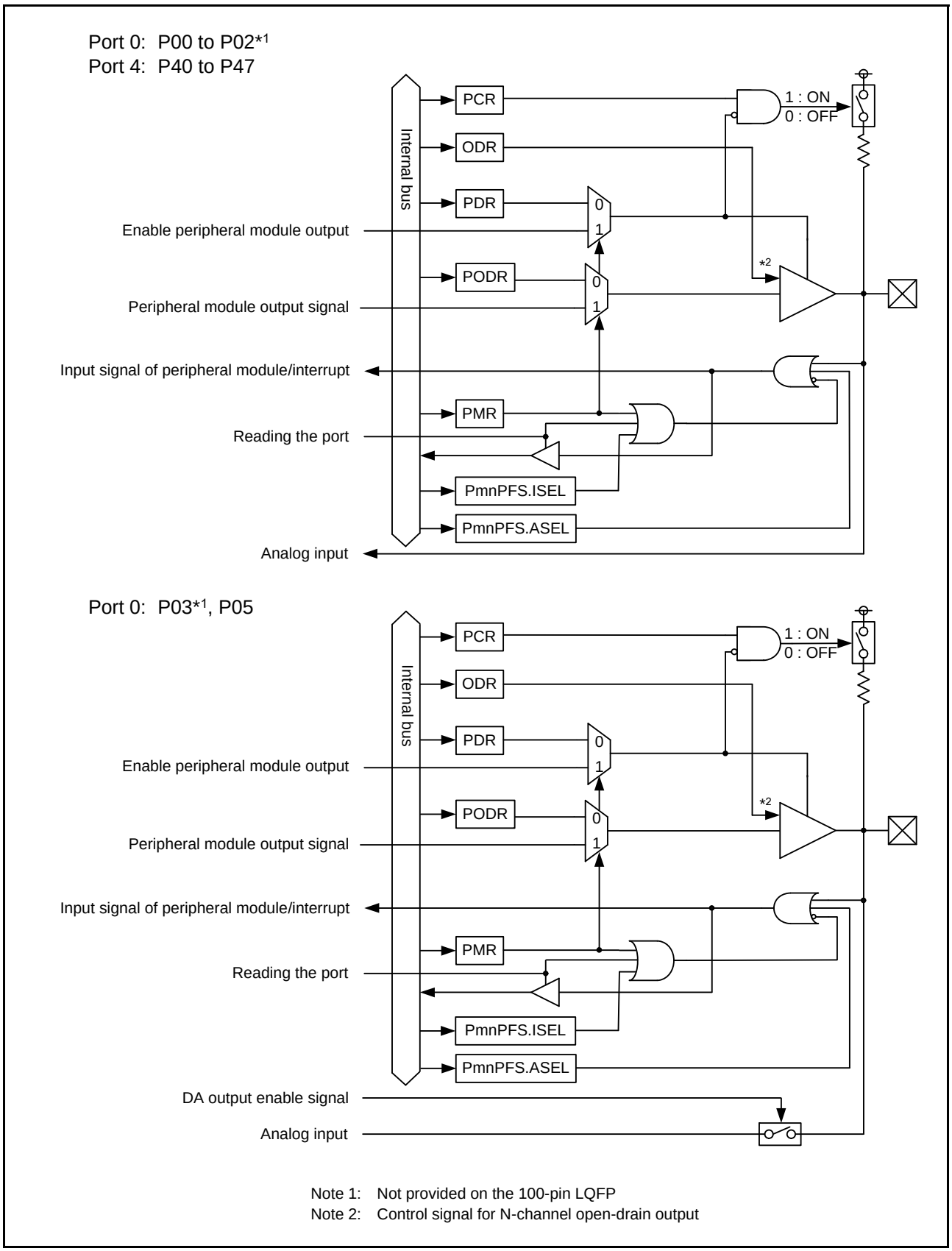
Table 20.2 Port Functions (1/2)

Port	Pin	Input Pull-up	Open Drain Output	Driving Abilify 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, P11	○	○	Fixed to high driving ability output	—
	P12 to P17	○	○	Fixed to high driving ability output	○
PORT2	P20, P21	○	○	Fixed to high driving ability output	○
	P22 to P25	○	○	Fixed to high driving ability output	—
	P26	○	○	Fixed to high driving ability output	—
	P27	○	○	○	—
PORT3	P30 to P33	○	○	Fixed to high driving ability output	○
	P34	○	○	Fixed to high driving ability output	—
	P35	—	—	—	—
	P36	○	○	Fixed to normal output	—
	P37	○	○	Fixed to high driving ability output	—
PORT4	P40 to P47	○	○	Fixed to normal output	—
PORT5	P50 to P52, P56, P57	○	○	○	○
	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 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	—

Table 20.2 Port Functions (2/2)

Port	Pin	Input Pull-up	Open Drain Output	Driving Abilify Switching	5-V Tolerant
PORTJ	PJ3, PJ5	○	○	Fixed to high driving ability output	—

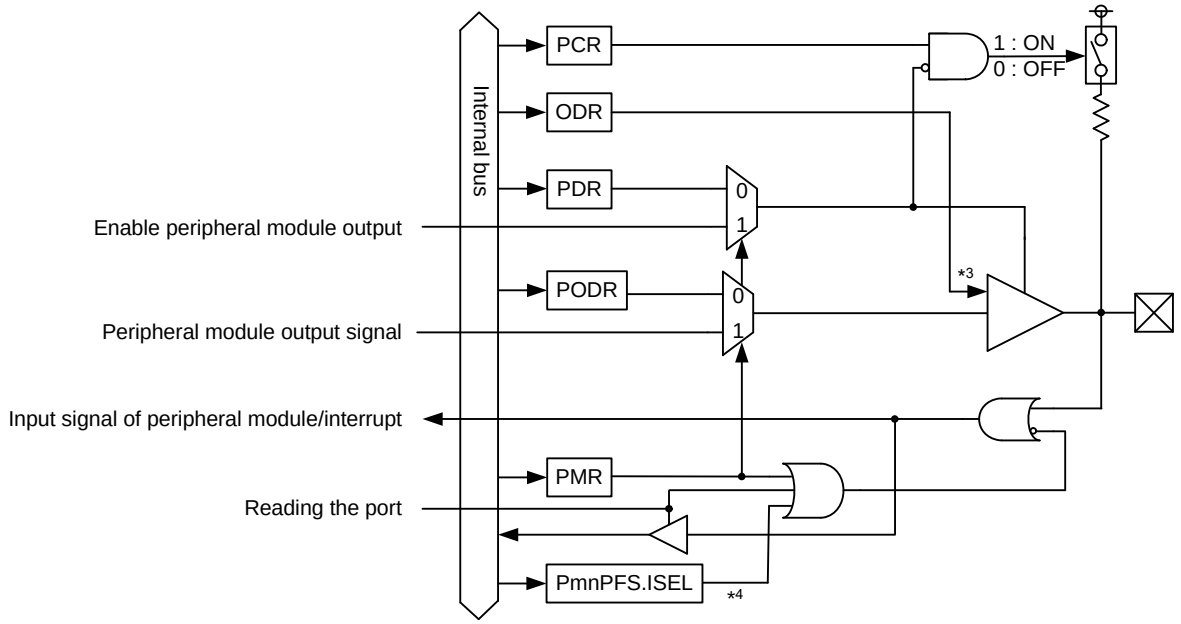
20.2 I/O Port Configuration



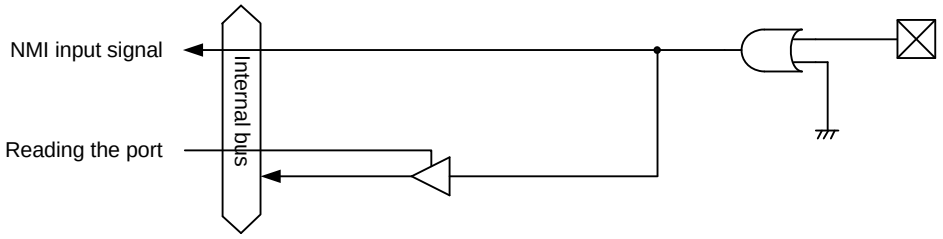
Note 1: Not provided on the 100-pin LQFP
Note 2: Control signal for N-channel open-drain output

Figure 20.1 I/O Port Configuration (1)

Port 0: P07
Port 1: P10*1, *2, P11*1, *2, P12 to P17
Port 2: P20 to P23
Port 3: P30 to P34, P36, P37
Port 7: P70*2
Port 8: P80 to P83*2, P84*1 to *2, P85*1 to *2, P86*2, P87*2
Port A: PA0 to PA6, PA7
Port F: PF0 to PF4*1, *2, PF5*2
Port J: PJ3, PJ5*2



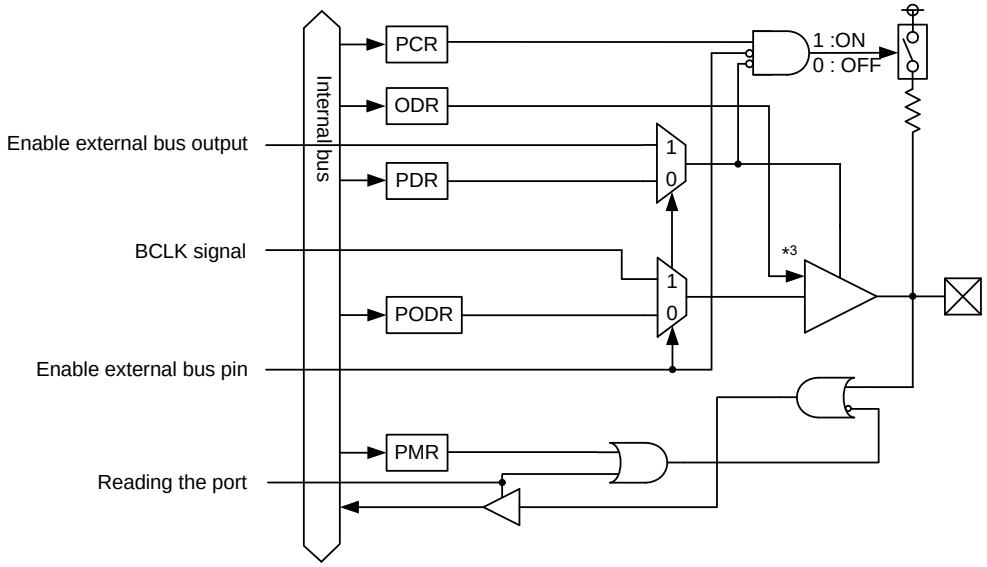
Port 3: P35



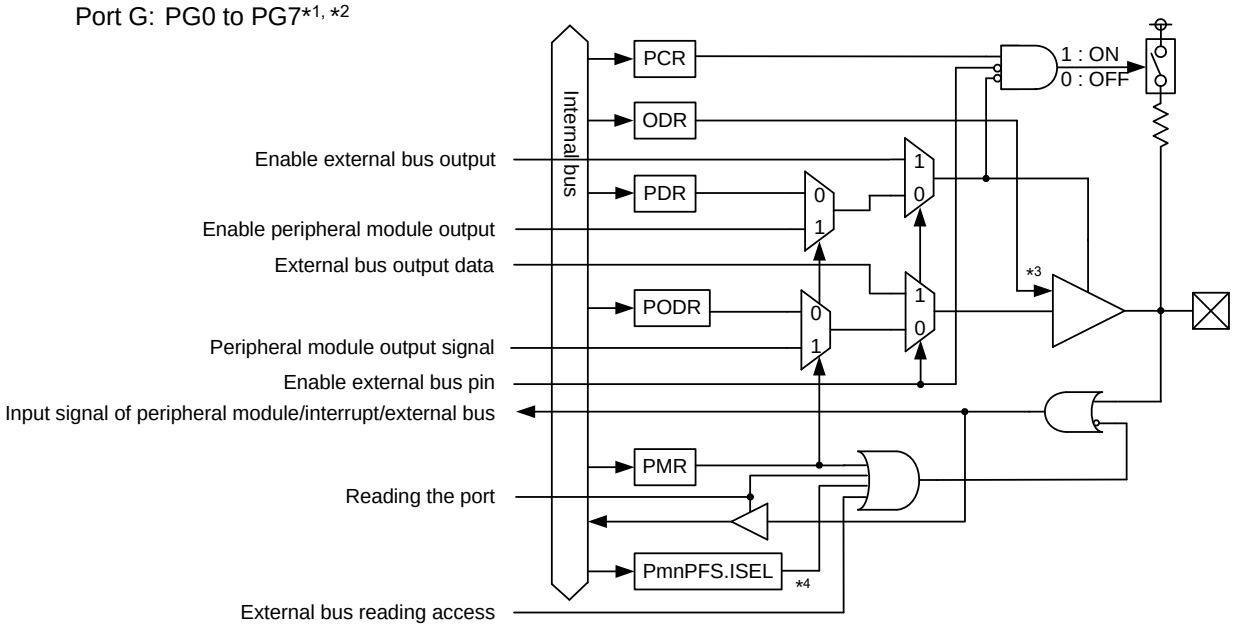
Note 1. Not provided on the 145-pin TFLGA and 144-pin LQFP
Note 2. Not provided on the 100-pin LQFP
Note 3. Control signal for N-channel open-drain output
Note 4. An external interrupt function is multiplexed on this pin

Figure 20.2 I/O Port Configuration (2)

Port 5: P53



Port 2: P24 to P27
Port 5: P50 to P52, P54, P55, P56*2, P57*1, *2
Port 6: P60 to P67*2
Port 7: P71 to P77*2
Port 9: P94 to P97*1, *2
Port A: PA0 to PA7
Port B: PB0 to PB7
Port C: PC0 to PC7
Port G: PG0 to PG7*1, *2



Note 1. Not provided on the 145-pin TFLGA and 144-pin LQFP
Note 2. Not provided on the 100-pin LQFP
Note 4. Control signal for N-channel open-drain output
Note 5. An external interrupt function is multiplexed on this pin

Figure 20.3 I/O Port Configuration (3)

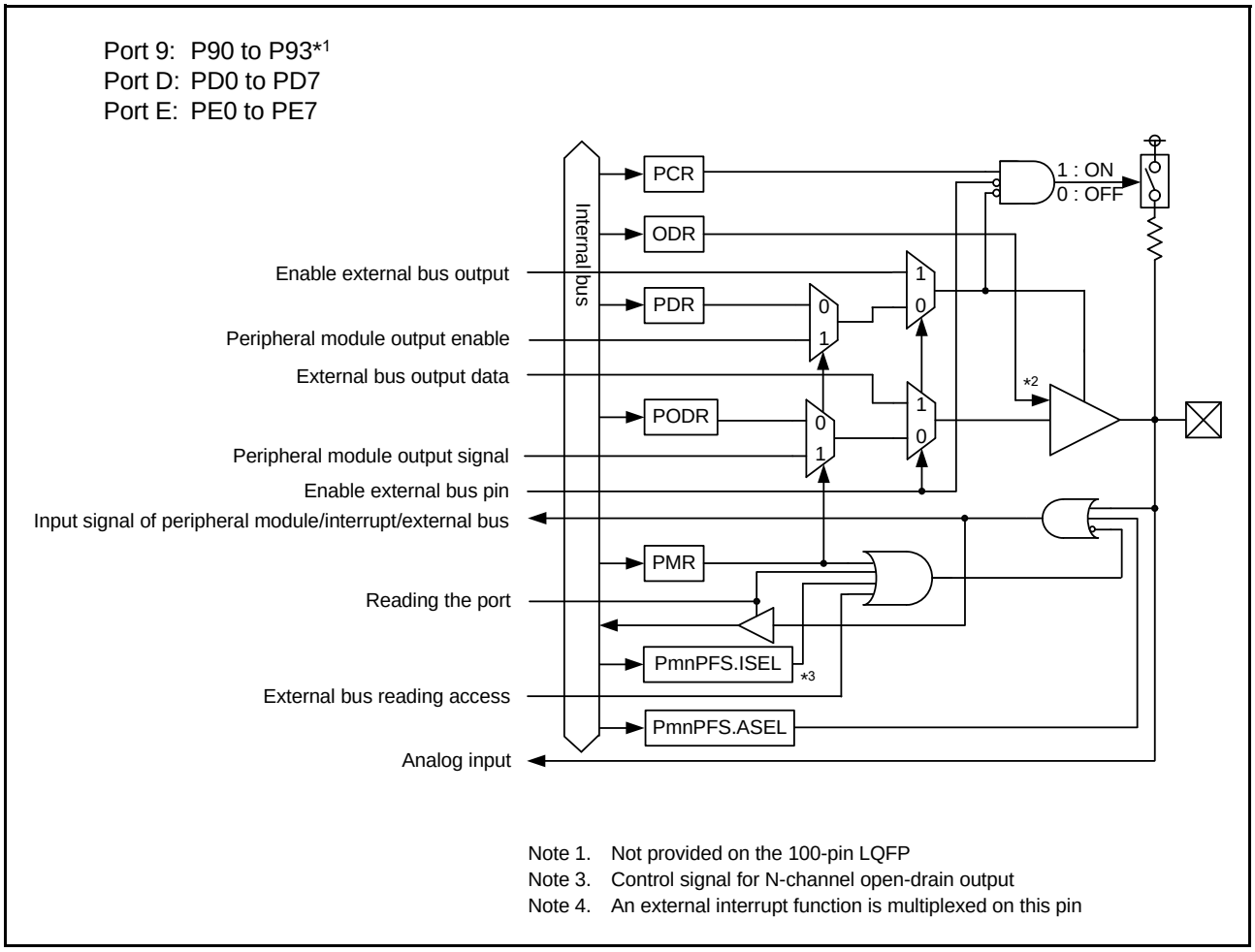


Figure 20.4 I/O Port Configuration (4)

20.3 Register Descriptions

20.3.1 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, 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 1 (output) to these bits.
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.

20.3.2 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.

20.3.3 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.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

20.3.4 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 functions.	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 are reserved. When writing, write 0 (general I/O port) to these bits.
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.

20.3.5 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	—	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 pins	R/W
b1	—	Reserved	Odd bit / Even bit	R/W
b2	B2	Pm1 Output Type Select	× 0: CMOS output	R/W
b3	—	Reserved	× 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 pins	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

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.
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.

20.3.6 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.
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.

20.3.7 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, a general port output pin, or a peripheral function 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 PORT3.PCR.B5 bit is reserved. 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.

20.3.8 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 (m = 0 to 5, A to E, H and J)	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 that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

20.4 Handling of Unused Pins

Details on the handling of unused pins are given in Table 20.3.

Table 20.3 Handling of Unused Pins

Pin Name	Handling
EMLE	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).
USB0_DP, USB1_DP	Keep these pins open.
USB0_DM, USB1_DM	
P35/NMI	Connect this pin to VCC via a resistor (pulling up).
EXTAL	Use this as a clock pin.
XTAL	Keep this pin open.
XCIN	Connect this pin to VCC or to VSS via a resistor (for pulling up or pulling down, respectively).
XCOUT	Keep this pin open.
Port 0 to Port 9, Port A to Port G, Port J	Connect each pin in turn to VCC or to VSS via a resistor (for pulling up or pulling down, respectively).
VREFH0	Connect this pin to AVCC0.
VREFL0	Connect this pin to AVSS0.
VREFH	Connect this pin to VCC.
VREFL	Connect this pin to VSS

21. Multi-Function Pin Controller (MPC)

21.1 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 21.1 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 21.1 Functions Assigned to Each Multiplexed Pin (1/16)

				Package		
Module/Function	**Channel**	**Pin Functions**	**Allocation Port**	**177-pin 176-pin**	**145-pin 144-pin**	**100-pin**
Interrupt		NMI(input)	P35	○	○	○
EXDMA controller	EXDMAC0	EDREQ0 (input)	P22	○	○	○
			P55	○	○	○
			P80	○	○	×
		EDACK0 (output)	P23	○	○	○
			P54	○	○	○
			P81	○	○	×
	EXDMAC1	EDREQ1 (input)	P24	○	○	○
			P57	○	×	×
			P82	○	○	×
		EDACK1 (output)	P25	○	○	○
			P56	○	○	×
			P83	○	○	×
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 21.1 Functions Assigned to Each Multiplexed Pin (2/16)

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 2	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 21.1 Functions Assigned to Each Multiplexed Pin (3/16)

				Package		
Module/Function	**Channel**	**Pin Functions**	**Allocation Port**	**177-pin 176-pin**	**145-pin 144-pin**	**100-pin**
Multi-function timer unit 2	MTU3	MTIOC3A (input/output)	P14	○	○	○
			P17	○	○	○
			PC1	○	○	○
			PC7	○	○	○
		MTIOC3B (input/output)	P17	○	○	○
			P22	○	○	○
			P80	○	○	×
			PB7	○	○	○
			PC5	○	○	○
		MTIOC3C (input/output)	P16	○	○	○
			P56	○	○	×
			PC0	○	○	○
			PC6	○	○	○
			PJ3	○	○	○
		MTIOC3D (input/output)	P16	○	○	○
			P23	○	○	○
			P81	○	○	×
			PB6	○	○	○
			PC4	○	○	○
	MTU4	MTIOC4A (input/output)	P24	○	○	○
			P82	○	○	×
			PA0	○	○	○
			PB3	○	○	○
			PE2	○	○	○
		MTIOC4B (input/output)	P30	○	○	○
			P54	○	○	○
			PC2	○	○	○
			PD1	○	○	○
			PE3	○	○	○
		MTIOC4C (input/output)	P25	○	○	○
			P83	○	○	×
			PB1	○	○	○
			PE1	○	○	○
			PE5	○	○	○
		MTIOC4D (input/output)	P31	○	○	○
			P55	○	○	○
			PC3	○	○	○
			PD2	○	○	○
			PE4	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (4/16)

				Package		
Module/Function	**Channel**	**Pin Functions**	**Allocation Port**	**177-pin 176-pin**	**145-pin 144-pin**	**100-pin**
Multi-function timer unit 2	MTU5	MTIC5U (input)	P12	○	○	○
			PA4	○	○	○
			PD7	○	○	○
		MTIC5V (input)	P11	○	×	×
			PA6	○	○	○
			PD6	○	○	○
		MTIC5W (input)	P10	○	×	×
			PB0	○	○	○
			PD5	○	○	○
	MTU	MTCLKA (input)	P14	○	○	○
			P24	○	○	○
			PA4	○	○	○
			PC6	○	○	○
		MTCLKB (input)	P15	○	○	○
			P25	○	○	○
			PA6	○	○	○
			PC7	○	○	○
		MTCLKC (input)	P22	○	○	○
			PA1	○	○	○
			PC4	○	○	○
		MTCLKD (input)	P23	○	○	○
			PA3	○	○	○
			PC5	○	○	○
Port output enable 2	POE0	POE0# (input)	PC4	○	○	○
			PD7	○	○	○
	POE1	POE1# (input)	PB5	○	○	○
			PD6	○	○	○
	POE2	POE2# (input)	P34	○	○	○
			PA6	○	○	○
			PD5	○	○	○
	POE3	POE3# (input)	P33	○	○	○
			PB3	○	○	○
			PD4	○	○	○
	POE8	POE8# (input)	P17	○	○	○
			P30	○	○	○
			PD3	○	○	○
			PE3	○	○	○
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	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (5/16)

				Package		
Module/Function	**Channel**	**Pin Functions**	**Allocation Port**	**177-pin 176-pin**	**145-pin 144-pin**	**100-pin**
16-bit timer pulse unit	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	○	○	○
	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	○	○	○
	TPU6	TIOCA6 (input/output)	PC6	○	○	×
		TIOCB6 (input/output)	PC7	○	○	×
		TIOCC6 (input/output)	PC4	○	○	×
		TIOCD6 (input/output)	PC5	○	○	×
	TPU7	TIOCA7 (input/output)	PD0	○	○	×
		TIOCB7 (input/output)	PD1	○	○	×
	TPU8	TIOCA8 (input/output)	PD2	○	○	×
		TIOCB8 (input/output)	PD3	○	○	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (6/16)

				Package		
Module/Function	**Channel**	**Pin Functions**	**Allocation Port**	**177-pin 176-pin**	**145-pin 144-pin**	**100-pin**
16-bit timer pulse unit	TPU9	TIOCA9 (input/output)	PE2	○	○	×
		TIOCB9 (input/output)	PE3	○	○	×
		TIOCC9 (input/output)	PE0	○	○	×
		TIOCD9 (input/output)	PE1	○	○	×
	TPU10	TIOCA10 (input/output)	PE4	○	○	×
		TIOCB10 (input/output)	PE5	○	○	×
	TPU11	TIOCA11 (input/output)	PE6	○	○	×
		TIOCB11 (input/output)	PE7	○	○	×
	TPU (unit 1)	TCLKE (input)	PC4	○	○	×
		TCLKF (input)	PC5	○	○	×
		TCLKG (input)	PD1	○	○	×
		TCLKH (input)	PD3	○	○	×
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	○	○	○
	PPG0	PO13 (output)	P13	○	○	○
			P15	○	○	○
		PO14 (output)	P16	○	○	○
		PO15 (output)	P14	○	○	○
			P17	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (7/16)

				Package		
Module/Function	**Channel**	**Pin Functions**	**Allocation Port**	**177-pin 176-pin**	**145-pin 144-pin**	**100-pin**
Programmable pulse generator	PPG1	PO16 (output)	P73	○	○	×
			PA0	○	○	○
		PO17 (output)	PA1	○	○	○
			PC0	○	○	○
		PO18 (output)	PA2	○	○	○
			PC1	○	○	○
			PE1	○	○	○
		PO19 (output)	P74	○	○	×
			PA3	○	○	○
		PO20 (output)	P75	○	○	×
			PA4	○	○	○
		PO21 (output)	PA5	○	○	○
			PC2	○	○	○
		PO22 (output)	P76	○	○	×
			PA6	○	○	○
	PPG1	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 21.1 Functions Assigned to Each Multiplexed Pin (8/16)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 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	○	○	○
Ethernet controller		REF50CK (input)	P76	○	○	×
			PB2	○	○	○
			PE5	○	○	○
		RMII_CRS_DV (input)	P83	○	○	×
			PB7	○	○	○
		RMII_TXD0 (output)	P81	○	○	×
			PB5	○	○	○
		RMII_TXD1 (output)	P82	○	○	×
			PB6	○	○	○
		RMII_RXD0 (input)	P75	○	○	×
			PB1	○	○	○
		RMII_RXD1 (input)	P74	○	○	×
			PB0	○	○	○
		RMII_TXD_EN (output)	P80	○	○	×
			PA0	○	○	○
			PB4	○	○	○
		RMII_RX_ER (input)	P77	○	○	×
			PB3	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (9/16)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Ethernet controller		ET_CRS (input)	P83	○	○	×
			PB7	○	○	○
		ET_RX_DV (input)	PC2	○	○	○
		ET_EXOUT (output)	P55	○	○	○
			PA6	○	○	○
		ET_LINKSTA (input)	P54	○	○	○
			PA5	○	○	○
		ET_ETXD0 (output)	P81	○	○	×
			PB5	○	○	○
		ET_ETXD1 (output)	P82	○	○	×
			PB6	○	○	○
		ET_ETXD2 (output)	PC5	○	○	○
		ET_ETXD3 (output)	PC6	○	○	○
		ET_ERXD0 (input)	P75	○	○	×
			PB1	○	○	○
		ET_ERXD1 (input)	P74	○	○	×
			PB0	○	○	○
		ET_ERXD2 (input)	PC1	○	○	○
			PE4	○	○	○
		ET_ERXD3 (input)	PC0	○	○	○
			PE3	○	○	○
		ET_TX_EN (output)	P80	○	○	×
			PA0	○	○	○
			PB4	○	○	○
		ET_TX_ER (output)	PC3	○	○	○
		ET_RX_ER (input)	P77	○	○	×
			PB3	○	○	○
		ET_TX_CLK (input)	PC4	○	○	○
		ET_RX_CLK (input)	P76	○	○	×
			PB2	○	○	○
			PE5	○	○	○
		ET_COL (input)	PC7	○	○	○
		ET_WOL (output)	P73	○	○	×
			PA1	○	○	○
			PA7	○	○	○
		ET_MDC (output)	P72	○	○	×
			PA4	○	○	○
		ET_MDIO (input/output)	P71	○	○	×
			PA3	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (10/16)

				Package		
Module/Function	**Channel**	**Pin Functions**	**Allocation Port**	**177-pin 176-pin**	**145-pin 144-pin**	**100-pin**
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	○	○	○
	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	○	○	○
	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	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (11/16)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Serial communications interface	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	○	○	○
	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	○	○	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (12/16)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Serial communications interface	SCI8	RXD8 (input)/ SMISO8 (input/output)/ SSCL8 (input/output)	PC6	○	○	○
		TXD8 (output)/ SMOSI8 (input/output)/ SSDA8 (input/output)	PC7	○	○	○
		SCK8 (input/output)	PC5	○	○	○
		CTS8# (input)/ RTS8# (output)/ SS8# (input)	PC4	○	○	○
	SCI9	RXD9 (input)/ SMISO9 (input/output)/ SSCL9 (input/output)	PB6	○	○	○
		TXD9 (output)/ SMOSI9 (input/output)/ SSDA9 (input/output)	PB7	○	○	○
		SCK9 (input/output)	PB5	○	○	○
		CTS9# (input)/ RTS9# (output)/ SS9# (input)	PB4	○	○	○
	SCI10	RXD10 (input)/ SMISO10 (input/output)/ SSCL10 (input/output)	P81	○	○	×
		TXD10 (output)/ SMOSI10 (input/output)/ SSDA10 (input/output)	P82	○	○	×
		SCK10 (input/output)	P80	○	○	×
		CTS10# (input)/ RTS10# (output)/ SS10# (input)	P83	○	○	×
	SCI11	RXD11 (input)/ SMISO11 (input/output)/ SSCL11 (input/output)	P76	○	○	×
		TXD11 (output)/ SMOSI11 (input/output)/ SSDA11 (input/output)	P77	○	○	×
		SCK11 (input/output)	P75	○	○	×
		CTS11# (input)/ RTS11# (output)/ SS11# (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 21.1 Functions Assigned to Each Multiplexed Pin (13/16)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
I^2C bus interface	RIIC0	SCL0[FM+] (input/output)	P12	○	○	○
		SDA0[FM+] (input/output)	P13	○	○	○
	RIIC1	SCL1 (input/output)	P21	○	○	×
		SDA1 (input/output)	P20	○	○	×
	RIIC2	SCL2-DS (input/output)	P16	○	○	○
		SDA2-DS	P17	○	○	○
	RIIC3	SCL3 (input/output)	PC0	○	○	×
		SDA3 (input/output)	PC1	○	○	×
USB 2.0 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	○	○	○
		USB0_ID (input)	P20	○	○	○
		USB0_DPUPE (output)	P14	○	○	○
			P23	○	○	○
			P31	○	○	○
		USB0_DPRPD (output)	P25	○	○	○
			P34	○	○	○
		USB0_DRPD (output)	P22	○	○	○
			P30	○	○	○
	USB1	USB1_VBUS (input)	P17	○	×	×
		USB1_DPUPE (output)	P15	○	×	×
CAN module	CAN0	CRX0 (input)*1	P33	○	○	○
			PD2	○	○	○
		CTX0 (output)*1	P32	○	○	○
			PD1	○	○	○
	CAN1	CRX1-DS (input)	P15	○	○	○
		CRX1 (input)	P55	○	○	○
		CTX1 (output)	P14	○	○	○
			P54	○	○	○
	CAN2	CRX2 (input)*1	P67	○	○	×
		CTX2 (output)*1	P66	○	○	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (14/16)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Serial peripheral interface	RSPI0	RSPCKA (input/output)	PA5	○	○	○
			PB0	○	○	○
			PC5	○	○	○
		MOSIA (input/output)	P16	○	○	○
			PA6	○	○	○
			PC6	○	○	○
		MISOA (input/output)	P17	○	○	○
			PA7	○	○	○
			PC7	○	○	○
		SSLA0 (input/output)	PA4	○	○	○
			PC4	○	○	○
		SSLA1 (output)	PA0	○	○	○
			PC0	○	○	○
		SSLA2 (output)	PA1	○	○	○
			PC1	○	○	○
		SSLA3 (output)	PA2	○	○	○
			PC2	○	○	○
	RSPI1	RSPCKB (input/output)	P27	○	○	○
			PE1	○	○	○
			PE5	○	○	○
		MOSIB (input/output)	P26	○	○	○
			PE2	○	○	○
			PE6	○	○	○
		MISOB (input/output)	P30	○	○	○
			PE3	○	○	○
			PE7	○	○	○
		SSLB0 (input/output)	P31	○	○	○
			PE4	○	○	○
		SSLB1 (output)	P50	○	○	○
			PE0	○	○	○
		SSLB2 (output)	P51	○	○	○
			PE1	○	○	○
		SSLB3 (output)	P52	○	○	○
			PE2	○	○	○
	RSPI2	RSPCKC (input/output)	PD3	○	○	×
		MOSIC (input/output)	PD1	○	○	×
		MISOC (input/output)	PD2	○	○	×
		SSLC0 (input/output)	PD4	○	○	×
		SSLC1 (output)	PD5	○	○	×
		SSLC2 (output)	PD6	○	○	×
		SSLC3 (output)	PD7	○	○	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (15/16)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
IEBus controller		IERXD (input)	P16	○	○	○
			PC2	○	○	○
		IETXD (output)	P17	○	○	○
			PC3	○	○	○
Realtime clock		RTCOUT (output)	P16	○	○	○
			P32	○	○	○
		RTCIC0 (input)*2	P30	○	○	○
		RTCIC1 (input)*2	P31	○	○	○
		RTCIC2 (input)*2	P32	○	○	○
12-bit A/D converter		AN000 (input)*2	P40	○	○	○
		AN001 (input)*2	P41	○	○	○
		AN002 (input)*2	P42	○	○	○
		AN003 (input)*2	P43	○	○	○
		AN004 (input)*2	P44	○	○	○
		AN005 (input)*2	P45	○	○	○
		AN006 (input)*2	P46	○	○	○
		AN007 (input)*2	P47	○	○	○
		AN008 (input)*2	PD0	○	○	○
		AN009 (input)*2	PD1	○	○	○
		AN010 (input)*2	PD2	○	○	○
		AN011 (input)*2	PD3	○	○	○
		AN012 (input)*2	PD4	○	○	○
		AN013 (input)*2	PD5	○	○	○
		AN014 (input)*2	P90	○	○	×
		AN015 (input)*2	P91	○	○	×
		AN016 (input)*2	P92	○	○	×
		AN017 (input)*2	P93	○	○	×
		AN018 (input)*2	P00	○	○	×
		AN019 (input)*2	P01	○	○	×
		AN020 (input)*2	P02	○	○	×
		ADTRG0# (input)	P07	○	○	○
			P16	○	○	○
			P25	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (16/16)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
10-bit A/D converter		AN0 (input)*2	PE2	○	○	○
		AN1 (input)*2	PE3	○	○	○
		AN2 (input)*2	PE4	○	○	○
		AN3 (input)*2	PE5	○	○	○
		AN4 (input)*2	PE6	○	○	○
		AN5 (input)*2	PE7	○	○	○
		AN6 (input)*2	PD6	○	○	○
		AN7 (input)*2	PD7	○	○	○
		ANEX0 (output)*2	PE0	○	○	○
		ANEX1 (input)*2	PE1	○	○	○
		ADTRG# (input)	P13	○	○	○
			P17	○	○	○
D/A converter		DA0 (output)*2	P03	○	○	×
		DA1 (output)*2	P05	○	○	○

Note 1. This pin is not provided in products less than 1 Mbyte of ROM.
Note 2. To make this pin available, set the pertinent pin for general port input (set the PORTm.PDR.Bn and PORTm.PMR.Bn bits to 0).

21.2 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.

21.2.1 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 always read as 0. The write value should always 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 (PFS Register Write Enable)

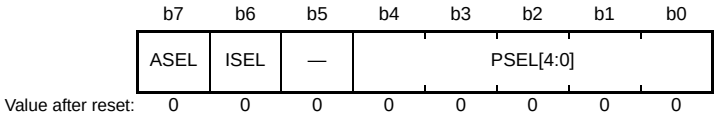
Writing to PmnPFS register is enabled only when the PFSWE bit is set to 1.
To set the PFSWE bit to 1, write 1 to the PFSWE bit after writing 0 to the B0WI bit.

B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

21.2.2 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
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.2.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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

The port mn pin function control register (PmnPFS) selects the pin function. Bits PSEL[4:0] 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 21.2 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

	Pin			
PSEL[4:0] Settings	**P00**	**P01**	**P02**	**P07**
00000b (initial value)	Hi-Z			
00101b	TMRI0	TMCI0	TMCI1	—
01001b	—	—	—	ADTRG0#
01010b	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6	—

—: Do not specify this value.

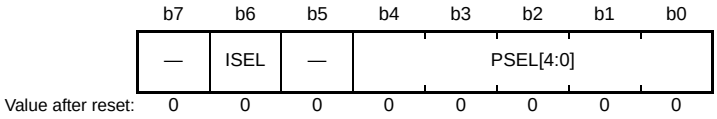
Table 21.3 Register Settings for Input/Output Pin Function in 100-Pin LQFP

	Pin
PSEL[4:0] Settings	**P07**
00000b (initial value)	Hi-Z
00101b	—
01001b	ADTRG0#
01010b	—

—: Do not specify this value.

21.2.3 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



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.4.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.4 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LQFP

PSEL[4:0] Settings	Pin							
	P10	P11	P12	P13	P14	P15	P16	P17
00000b (initial value)	Hi-Z							
00001b	MTIC5W	MTIC5V	MTIC5U	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	—	—	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	—	—	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
00100b	—	—	—	—	TCLKA	TCLKB	TCLKC	TCLKD
00101b	TMRI3	TMCI3	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
00110b	—	—	—	PO13	PO15	PO13	PO14	PO15
00111b	—	—	—	—	—	—	RTCOUT	POE8#
01001b	—	—	—	ADTRG#	—	—	ADTRG0#	ADTRG#
01010b	—	SCK2	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
01101b	—	—	—	—	—	—	MOSIA	MISOA
01111b	—	—	SCL0[FM+]	SDA0[FM+]	—	—	SCL2-DS	SDA2-DS
10000b	—	—	—	—	CTX1	CRX1-DS	IERXD	IETXD
10001b	—	—	—	—	USB0_DPUPE	USB1_DPUPE	USB0_VBUS	USB1_VBUS
10010b	—	—	—	—	USB0_OVRCURA	—	USB0_VBUSEN	—
100011b	—	—	—	—	—	—	USB0_OVRCURB	—

—: Do not specify this value.

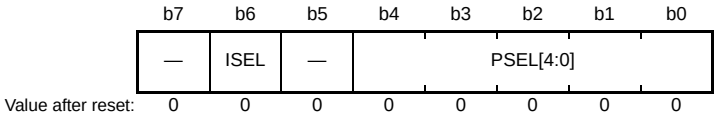
Table 21.5 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-/100-Pin LQFP

PSEL[4:0] Settings	Pin					
	P12	P13	P14	P15	P16	P17
00000b (initial value)	Hi-Z					
00001b	—	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
00100b	—	—	TCLKA	TCLKB	TCLKC	TCLKD
00101b	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
00110b	—	PO13	PO15	PO13	PO14	PO15
00111b	—	—	—	—	RTCOUT	POE8#
01001b	—	ADTRG#	—	—	ADTRG0#	ADTRG#
01010b	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
01101b	—	—	—	—	MOSIA	MISOA
01111b	SCL0[FM+]	SDA0[FM+]	—	—	SCL2-DS	SDA2-DS
10000b	—	—	CTX1	CRX1-DS	IERXD	IETXD
10001b	—	—	USB0_DPUPE	—	USB0_VBUS	—
10010b	—	—	USB0_OVRCURA	—	USB0_VBUSEN	—
100011b	—	—	—	—	USB0_OVRCURB	—

—: Do not specify this value.

21.2.4 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
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.6.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.6 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin							
	P20	P21	P22	P23	P24	P25	P26	P27
00000b (initial value)	Hi-Z							
00001b	MTIOC1A	MTIOC1B	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC2A	MTIOC2B
00010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB	—	—
00011b	TIOCB3	TIOCA3	TIOCC3	TIOCD3	TIOCB4	TIOCA4	—	—
00101b	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO1	TMCI3
00110b	PO0	PO1	PO2	PO3	PO4	PO5	PO6	PO7
01001b	—	—	—	—	—	ADTRG0#	—	—
01010b	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0	TXD3 SMOSI3 SSDA3	SCK3	RXD3 SMISO3 SSCL3	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	—	CTS0# RTS0# SS0#	—	—	CTS3# RTS3# SS3#	—
01101b	—	—	—	—	—	—	MOSIB	RSPCKB
01111b	SDA1	SCL1	—	—	—	—	—	—
10011b	USB0_ID	USB0_EXICEN	USB0_DRPD	USB0_DPUPE	USB0_VBUSEN	USB0_DPRPD	—	—
11000b	—	—	EDREQ0	EDACK0	EDREQ1	EDACK1	—	—

—: Do not specify this value.

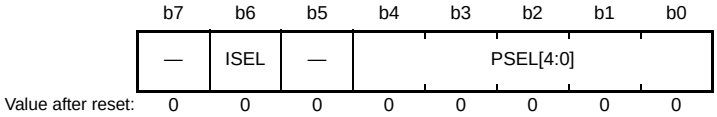
Table 21.7 Register Settings for Input/Output Pin Function in 100-Pin LQFP

PSEL[4:0] Settings	Pin							
	P20	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**
00000b (initial value)	Hi-Z							
00001b	MTIOC1A	MTIOC1B	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC2A	MTIOC2B
00010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB	—	—
00011b	TIOCB3	TIOCA3	TIOCC3	TIOCD3	TIOCB4	TIOCA4	—	—
00101b	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO1	TMCI3
00110b	PO0	PO1	PO2	PO3	PO4	PO5	PO6	PO7
01001b	—	—	—	—	—	ADTRG0#	—	—
01010b	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0	TXD3 SMOSI3 SSDA3	SCK3	RXD3 SMISO3 SSCL3	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	—	CTS0# RTS0# SS0#	—	—	CTS3# RTS3# SS3#	—
01101b	—	—	—	—	—	—	MOSIB	RSPCKB
10011b	USB0_ID	USB0_EXICEN	USB0_DRPD	USB0_DPUPE	USB0_VBUSEN	USB0_DPRPD	—	—
11000b	—	—	EDREQ0	EDACK0	EDREQ1	EDACK1	—	—

—: Do not specify this value.

21.2.5 Port 3n 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



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.8.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.8 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LQFP

PSEL[4:0] Settings	Pin				
	P30	P31	P32	P33	P34
00000b (initial value)	Hi-Z				
00001b	MTIOC4B	MTIOC4D	MTIOC0C	MTIOC0D	MTIOC0A
00011b	—	—	TIOCC0	TIOCD0	—
00101b	TMRI3	TMCI2	TMO3	TMRI3	TMCI3
00110b	PO8	PO9	PO10	PO11	PO12
00111b	POE8#	—	RTCOUT	POE3#	POE2#
01010b	RXD1 SMISO1 SSCL1	—	TXD6 SRXD6 SSDA6	RXD6 SMISO6 SSCL6	SCK6
01011b	—	CTS1# RTS1# SS1#	TXD0 SMOSI6 SSDA0	RXD0 SMISO0 SSCL0	SCK0
01101b	MISOB	SSLB0	—	—	—
10000b	—	—	CTX0	CRX0	—
10011b	USB0_DRPD	USB0_DPUPE	USB0_VBUSEN	—	USB0_DPRPD

—: Do not specify this value.

21.2.6 P4n Pin Function Control Registers (P4nPFS) (n = 0 to 7)

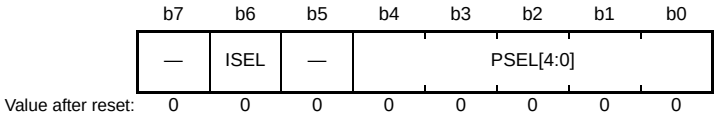
Address(es): P40PFS 0008 C160h, P41PFS 0008 C161h, P42PFS 0008 C162h, P43PFS 0008 C163h
P44PFS 0008 C164h, P45PFS 0008 C165h, P46PFS 0008 C166h, P47PFS 0008 C167h

	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 always read as 0. The write value should always 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

21.2.7 P5n Pin Function Control Registers (P5nPFS) (n = 0 to 2, 4, to 7)

Address(es): P50PFS 0008 C168h, P51PFS 0008 C169h, P52PFS 0008 C16Ah
P54PFS 0008 C16Ch, P55PFS 0008 C16Dh, P56PFS 0008 C16Eh, P57PFS 0008 C16Fh



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.9.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.9 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LQFP

PSEL[4:0] Settings	Pin						
	P50	P51	P52	P54	P55	P56	P57
00000b (initial value)	Hi-Z						
00001b	—	—	—	MTIOC4B	MTIOC4D	MTIOC3C	—
00011b	—	—	—	—	—	TIOCA1	—
00101b	—	—	—	TMCI1	TMO3	—	—
01010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	—	—
01011b	—	—	—	CTS2# RTS2# SS2#	—	—	—
01101b	SSLB1	SSLB2	SSLB3	—	—	—	—
10000b	—	—	—	CTX1	CRX1	—	—
10001b	—	—	—	ET_LINKSTA	ET_EXOUT	—	—
11000b	—	—	—	EDACK0	EDREQ0	EDACK1	EDREQ1

—: Do not specify this value.

Table 21.10 Register Settings for Input/Output Pin Function in 144-Pin LQFP, 145-Pin TFLGA

PSEL[4:0] Settings	Pin					
	P50	**P51**	**P52**	**P54**	**P55**	**P56**
00000b (initial value)	Hi-Z					
00001b	—	—	—	MTIOC4B	MTIOC4D	MTIOC3C
00011b	—	—	—	—	—	TIOCA1
00101b	—	—	—	TMCI1	TMO3	—
01010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	—
01011b	—	—	—	CTS2# RTS2# SS2#	—	—
01101b	SSLB1	SSLB2	SSLB3	—	—	—
10000b	—	—	—	CTX1	CRX1	—
10001b	—	—	—	ET_LINKSTA	ET_EXOUT	—
11000b	—	—	—	EDACK0	EDREQ0	EDACK1

—: Do not specify this value.

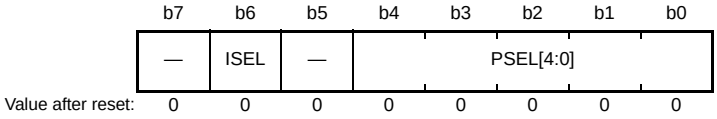
Table 21.11 Register Settings for Input/Output Pin Function in 100-Pin LQFP

PSEL[4:0] Settings	Pin				
	P50	**P51**	**P52**	**P54**	**P55**
00000b (initial value)	Hi-Z				
00001b	—	—	—	MTIOC4B	MTIOC4D
00101b	—	—	—	TMCI1	TMO3
01010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—
01011b	—	—	—	CTS2# RTS2# SS2#	—
01101b	SSLB1	SSLB2	SSLB3	—	—
10000b	—	—	—	CTX1	CRX1
10001b	—	—	—	ET_LINKSTA	ET_EXOUT
11000b	—	—	—	EDACK0	EDREQ0

—: Do not specify this value.

21.2.8 P6n Pin Function Control Registers (P6nPFS) (n = 0, 1, 6, 7)

Address(es): P60PFS 0008 C170h, P61PFS 0008 C171h, P66PFS 0008 C176h, P67PFS 0008 C177h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.12.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

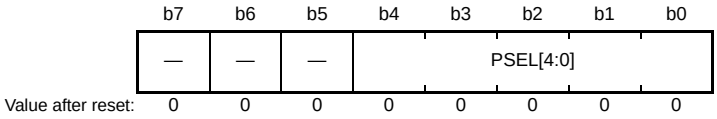
Table 21.12 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

	Pin			
PSEL[4:0] Settings	**P60**	**P61**	**P66**	**P67**
00000b (initial value)			Hi-Z	
01010b	SCK9	—	—	—
01011b	—	CTS9# RTS9# SS9#	—	—
10000b	—	—	CTX2	CRX2

—: Do not specify this value.

21.2.9 P7n Pin Function Control Registers (P7nPFS) (n = 0 to 7)

Address(es): P70PFS 0008 C178h, 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
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.13.	R/W
b7 to b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

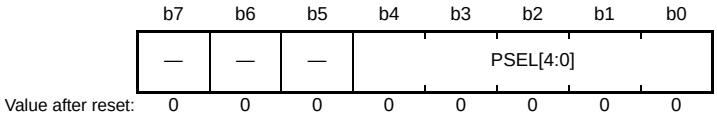
Table 21.13 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin							
	P70	P71	P72	P73	P74	P75	P76	P77
00000b (initial value)	Hi-Z							
00110b	—	—	—	PO16	PO19	PO20	PO22	PO23
01010b	SCK4	—	—	—	—	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
01011b	—	—	—	—	CTS11# RTS11 SS11#	—	—	—
10001b	—	ET_MDIO	ET_MDC	ET_WOL	ET_ERXD1	ET_ERXD0	ET_RX_CLK	ET_RX_ER
10010b	—	—	—	—	RMII_RXD1	RMII_RXD0	REF50CK	RMII_RX_ER

—: Do not specify this value.

21.2.10 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
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.14.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

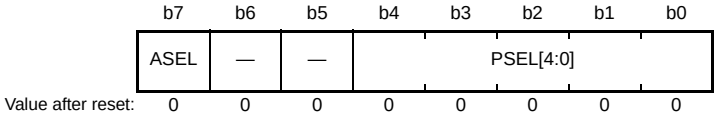
Table 21.14 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin					
	P80	P81	P82	P83	P86	P87
00000b (initial value)	Hi-Z					
00001b	MTIOC3B	MTIOC3D	MTIOC4A	MTIC4C	—	—
00011b	—	—	—	—	TIOCA0	TIOCA2
00110b	PO26	PO27	PO28	—	—	—
01010b	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10	—	—	—
01011b	—	—	—	CTS10# RTS10# SS10#	—	—
10001b	ET_TX_EN	ET_ETXD0	ET_ETXD1	ET_CRS	—	—
10010b	RMII_TXD_EN	RMII_TXD0	RMII_TXD1	RMII_CRS_DV	—	—
11000b	EDREQ0	EDACK0	EDREQ1	EDACK1	—	—

—: Do not specify this value.

21.2.11 P9n Pin Function Control Registers (P9nPFS) (n = 0 to 3)

Address(es): P90PFS 0008 C188h, P91PFS 0008 C189h, P92PFS 0008 C18Ah, P93PFS 0008 C18Bh



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.15.	R/W
b6 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

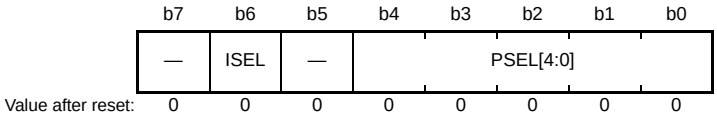
Table 21.15 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin			
	P90	P91	P92	P93
00000b (initial value)	Hi-Z			
01010b	TXD7 SMOSI7 SSDA7	SCK7	RXD7 SMISO7 SSCL7	—
01011b	—	—	—	CTS7# RTS7# SS7#

—: Do not specify this value.

21.2.12 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
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.16.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

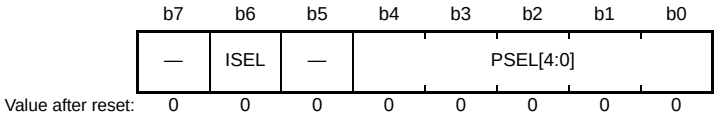
Table 21.16 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LQFP

PSEL[4:0] Settings	Pin							
	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7
00000b (initial value)	Hi-Z							
00001b	MTIOC4A	MTIOC0B	—	MTIOC0D	MTIC5U	—	MTIC5V	—
00010b	—	MTCLKC	—	MTCLKD	MTCLKA	—	MTCLKB	—
00011b	TIOCA0	TIOCB0	—	TIOCD0	TIOCA1	TIOCB1	TIOCA2	TIOCB2
00100b	—	—	—	TCLKB	—	—	—	—
00101b	—	—	—	—	TMRI0	—	TMCI3	—
00110b	PO16	PO17	PO18	PO19	PO20	PO21	PO22	PO23
00111b	—	—	—	—	—	—	POE2#	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—	—
01011b	—	—	—	—	—	—	CTS5# RTS5# SS5#	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MOSOA
10001b	ET_TX_EN	ET_WOL	—	ET_MDIO	ET_MDC	ET_LINKSTA	ET_EXOUT	ET_WOL
10010b	RMII_TXD_EN	—	—	—	—	—	—	—

—: Do not specify this value.

21.2.13 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



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.17.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.17 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin							
	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
00000b (initial value)					Hi-Z			
00001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
00010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
00011b	TIOCA3	TIOCB3	TIOCC3	TIOCD3	TIOCA4	TIOCB4	TIOCA5	TIOCB5
00100b	—	—	TCLKC	TCLKD	—	—	—	—
00101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
00110b	PO24	PO25	PO26	PO27	PO28	PO29	PO30	PO31
00111b	—	—	—	POE3#	—	POE1#	—	—
01010b	RXD4 SMISO4 SSCL4	TX4D SMOSI4 SSDA4	CTS4# RTS4# SS4#	SCK4	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
01101b	RSPCKA	—	—	—	—	—	—	—
10001b	ET_ERXD1	ET_ERXD0	ET_RX_CLK	ET_RX_ER	ET_TX_EN	ET_ETXD0	ET_ETXD1	ET_CRS
10010b	RMII_RXD1	RMII_RXD0	REF50CK	RMII_RX_ER	RMII_TXD_EN	RMII_TXD0	RMII_TXD1	RMII_CRS_DV

—: Do not specify this value.

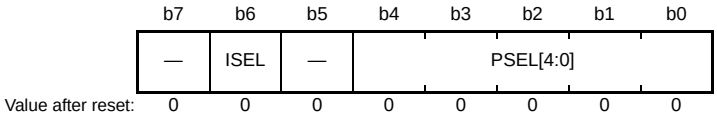
Table 21.18 Register Settings for Input/Output Pin Function in 100-Pin LQFP

PSEL[4:0] Settings	Pin							
	PB0	**PB1**	**PB2**	**PB3**	**PB4**	**PB5**	**PB6**	**PB7**
00000b (initial value)					Hi-Z			
00001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
00010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
00011b	TIOCA3	TIOCB3	TIOCC3	TIOCD3	TIOCA4	TIOCB4	TIOCA5	TIOCB5
00100b	—	—	TCLKC	TCLKD	—	—	—	—
00101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
00110b	PO24	PO25	PO26	PO27	PO28	PO29	PO30	PO31
00111b	—	—	—	POE3#	—	POE1#	—	—
01010b	—	—	—	—	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
01101b	RSPCKA	—	—	—	—	—	—	—
10001b	ET_ERXD1	ET_ERXD0	ET_RX_CLK	ET_RX_ER	ET_TX_EN	ET_ETXD0	ET_ETXD1	ET_CRS
10010b	RMII_RXD1	RMII_RXD0	REF50CK	RMII_RX_ER	RMII_TXD_EN	RMII_TXD0	RMII_TXD1	RMII_CRS_DV

—: Do not specify this value.

21.2.14 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



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.19.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.19 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin							
	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
00000b (initial value)	Hi-Z							
00001b	MTIOC3C	MTIOC3A	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
00010b	—	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00011b	TCLKC	TCLKD	TCLKA	TCLKB	TIOCC6	TIOCD6	TIOCA6	TIOCB6
00100b	—	—	—	—	TCLKE	TCLKF	—	—
00101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
00110b	PO17	PO18	PO21	PO24	PO25	PO29	PO30	PO31
00111b	—	—	—	—	POE0#	—	—	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	CTS5# RTS5# SS5#	—	—	—	CTS8# RTS8# SS8#	—	—	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
01111b	SCL3	SDA3	—	—	—	—	—	—
10000b	—	—	IERXD	IETXD	—	—	—	—
10001b	ET_ERXD3	ET_ERXD2	ET_RX_DV	ET_TX_ER	ET_TX_CLK	ET_ETXD2	ET_ETXD3	ET_COL

—: Do not specify this value.

Table 21.20 Register Settings for Input/Output Pin Function in 100-Pin LQFP

PSEL[4:0] Settings	Pin							
	PC0	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**
00000b (initial value)	Hi-Z							
00001b	MTIOC3C	MTIOC3A	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
00010b	—	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00011b	TCLKC	TCLKD	TCLKA	TCLKB	—	—	—	—
00101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
00110b	PO17	PO18	PO21	PO24	PO25	PO29	PO30	PO31
00111b	—	—	—	—	POE0#	—	—	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	CTS5# RTS5# SS5#	—	—	—	CTS8# RTS8# SS8#	—	—	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
10000b	—	—	IERXD	IETXD	—	—	—	—
10001b	ET_ERXD3	ET_ERXD2	ET_RX_DV	ET_TX_ER	ET_TX_CLK	ET_ETXD2	ET_ETXD3	ET_COL

—: Do not specify this value.

21.2.15 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[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.21.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.21 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin							
	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7
00000b (initial value)					Hi-Z			
00001b	—	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
00011b	TIOCA7	TIOCB7	TIOCA8	TIOCB8	—	—	—	—
00100b	—	TCLKG	—	TCLKH	—	—	—	—
00111b	—	—	—	POE8#	POE3#	POE2#	POE1#	POE0#
01101b	—	MOSIC	MISOC	RSPCKC	SSLC0	SSLC1	SSLC2	SSLC3
10000b	—	CTX0	CRX0	—	—	—	—	—

—: Do not specify this value.

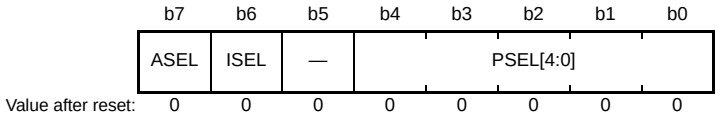
Table 21.22 Register Settings for Input/Output Pin Function in 100-Pin LQFP

PSEL[4:0] Settings	Pin						
	PD1	PD2	PD3	PD4	PD5	PD6	PD7
00000b (initial value)				Hi-Z			
00001b	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
00111b	—	—	POE8#	POE3#	POE2#	POE1#	POE0#
10000b	CTX0	CRX0	—	—	—	—	—

—: Do not specify this value.

21.2.16 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



Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.23.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.23 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin							
	PE0	PE1	PE2	PE3	PE4	PE5	PE6	PE7
00000b (initial value)	Hi-Z							
00001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C	—	—
00010b	—	—	—	—	MTIOC1A	MTIOC2B	—	—
00011b	TIOCC9	TIOCD9	TIOCA9	TIOCB9	TIOCA10	TIOCB10	TIOCA11	TIOCB11
00110b	—	PO18	PO23	PO26	PO28	—	—	—
00111b	—	—	—	POE8#	—	—	—	—
01011b	—	—	—	—	—	—	CTS4# RTS4# SS4#	—
01100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSSL12 RXDX12	CTS12# RTS12# SS12#	—	—	—	—
01101b	SSLB1	SSLB2	SSLB3	MISOB	SSLB0	RSPCKB	MOSIB	MISOB
01110b	—	RSPCKB	MOSIB	—	—	—	—	—
10001b	—	—	—	ET_ERXD3	ET_ERXD2	ET_RX_CLK	—	—
10010b	—	—	—	—	—	REF50CK	—	—

—: Do not specify this value.

Table 21.24 Register Settings for Input/Output Pin Function in 100-Pin LQFP

PSEL[4:0] Settings	Pin							
	PE0	**PE1**	**PE2**	**PE3**	**PE4**	**PE5**	**PE6**	**PE7**
00000b (initial value)	Hi-Z							
00001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C	—	—
00010b	—	—	—	—	MTIOC1A	MTIOC2B	—	—
00110b	—	PO18	PO23	PO26	PO28	—	—	—
00111b	—	—	—	POE8#	—	—	—	—
01100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSSL12 RXDX12	CTS12# RTS12# SS12#	—	—	—	—
01101b	SSLB1	SSLB2	SSLB3	MISOB	SSLB0	RSPCKB	MOSIB	MISOB
01110b	—	RSPCKB	MOSIB	—	—	—	—	—
10001b	—	—	—	ET_ERXD3	ET_ERXD2	ET_RX_CLK	—	—
10010b	—	—	—	—	—	REF50CK	—	—

—: Do not specify this value.

21.2.17 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

Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.25.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always 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	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

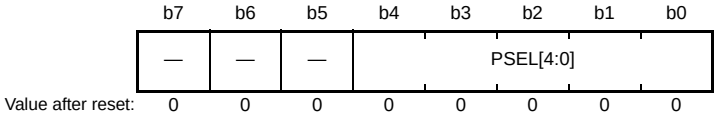
Table 21.25 Register Settings for Input/Output Pin Function in 176-Pin LQFP, 177-Pin TFLGA, 176-Pin LFBGA

PSEL[4:0] Settings	Pin			
	PF0	PF1	PF2	PF5
00000b (initial value)	Hi-Z			
01010b	TXD1 SMOSI1 SSDA1	SCK1	RXD1 SMISO1 SSCL1	—

—: Do not specify this value.

21.2.18 PJ3 Pin Function Control Register (PJ3PFS)

Address(es): PJ3PFS 0008 C1D3h



Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.26.	R/W
b7-b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Table 21.26 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LQFP

PSEL[4:0] Settings	Pin
	PJ3
00000b (initial value)	Hi-Z
00001b	MTIOC3C
01010b	CTS6# RTS6# SS6#
01011b	CTS0# RTS0# SS0#

21.2.19 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

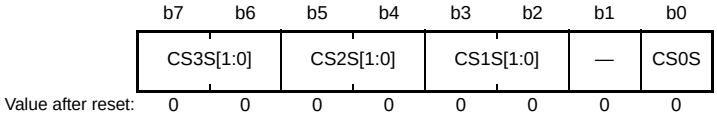
CSnE Bits (CSn Enable) (n = 0 to 7)

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.

21.2.20 CS Output Pin Select Register 0 (PFCSS0)

Address(es): 0008 C102h



Bit	Symbol	Bit Name	Description	R/W
b0	CS0S	CS0# Output Pin Select	0: Set P60 as CS0# output pin 1: Set PC7 as CS0# output pin	R/W
b1	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b3, b2	CS1S[1:0]	CS1# Output Pin Select	b3 b2 0 0: Set P61 as CS1# output pin 0 1: Set P71 as CS1# output pin 0 X: Set PC6 as CS1# output pin	R/W
b5, b4	CS2S[1:0]	CS2# Output Pin Select	b5 b4 0 0: Set P62 as CS2# output pin 0 1: Set P72 as CS2# output pin 0 X: Set PC5 as CS2# output pin	R/W
b7, b6	CS3S[1:0]	CS3# Output Pin Select	b7 b6 0 0: Set P63 as CS3# output pin 0 1: Set P73 as CS3# output pin 0 X: Set PC4 as CS3# output pin	R/W

X : Don't care

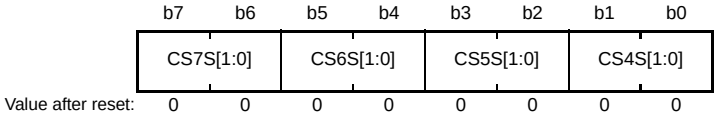
CS0S Bit (CS0# Output Pin Select)

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.

21.2.21 CS Output Pin Select Register 1 (PFCSS1)

Address(es): 0008 C103h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	CS4S[1:0]	CS4# Output Pin Select	b1 b0 0 0: Set P64 as CS4# output pin 0 1: Set P74 as CS4# output pin 0 X: Set P24 as CS4# output pin	R/W
b3, b2	CS5S[1:0]	CS5# Output Pin Select	b3 b2 0 0: Set P65 as CS5# output pin 0 1: Set P75 as CS5# output pin 0 X: Set P25 as CS5# output pin	R/W
b5, b4	CS6S[1:0]	CS6# Output Pin Select	b5 b4 0 0: Set P66 as CS6# output pin 0 1: Set P76 as CS6# output pin 0 X: Set P26 as CS6# output pin	R/W
b7, b6	CS7S[1:0]	CS7# Output Pin Select	b7 b6 0 0: Set P67 as CS7# output pin 0 1: Set P77 as CS7# output pin 0 X: Set P27 as CS7# output pin	R/W

X : Don't care

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.

21.2.22 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

21.2.23 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	AddressA16 Output Enable	0: DisablesA16 output. 1: EnablesA16 output.	R/W
b1	A17E	AddressA17 Output Enable	0: DisablesA17 output. 1: EnablesA17 output.	R/W
b2	A18E	AddressA18 Output Enable	0: DisablesA18 output. 1: EnablesA18 output.	R/W
b3	A19E	AddressA19 Output Enable	0: DisablesA19 output. 1: EnablesA19 output.	R/W
b4	A20E	AddressA20 Output Enable	0: DisablesA20 output. 1: EnablesA20 output.	R/W
b5	A21E	AddressA21 Output Enable	0: DisablesA21 output. 1: EnablesA21 output.	R/W
b6	A22E	AddressA22 Output Enable	0: DisablesA22 output. 1: EnablesA22 output.	R/W
b7	A23E	AddressA23 Output Enable	0: DisablesA23 output. 1: EnablesA23 output.	R/W

21.2.24 External Bus Control Register 0 (PFBCR0)

Address(es): 0008 C106h

	b7	b6	b5	b4	b3	b2	b1	b0
	WR32BC32E	WR1BC1E	DH32E	DHE	—	—	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	0: Configures PC0 to PC7 as the external address bus A16 to A23. 1: Configures P90 to P97 as the external address bus A16 to A23.	R/W
b3, b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	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 P56 and P57 as I/O port 1: Set P56 as WR#2 or BC2#, and set P57 as WR#3 or BC3#	R/W

21.2.25 External Bus Control Register 1 (PFBCR1)

Address(es): 0008 C107h

	b7	b6	b5	b4	b3	b2	b1	b0
	SDCLKE	DQM1E	—	MDSDE	—	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: Configures P57 as the WAIT# input pin. 0 1: Configures P55 as the WAIT# input pin. 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: Configures P54 as an I/O port pin. 1: Configures P54 as the ALE pin.	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b4	MDSDE	SDRAM Pin Enable (CKE, SDCS#, RAS#, CAS#, WE#, DQM0 outputs)	0: Disables CKE,SDCS#, RAS#, CAS#, WE#, and DQM0 outputs. 1: Enables CKE,SDCS#, RAS#, CAS#, WE#, and DQM0 outputs.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	DQM1E	DQM1 Enable	b4 b6 0 x: Disables DQM1 output 1 0: Disables DQM1 output 1 1: Enables DQM1 output	R/W
b7	SDCLKE	SDCLK Enable	0: Disables SDCLK output 1: Enables SDCLK output	R/W

X : Don’t care

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.

MDSDE Bit (SDRAM Pin Enable)

This bit enables or disables output of the SDRAM pin (CKE,SDCS#, RAS#, CAS#, WE#, 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 cleared to 0, setting of the DQM1E bit has no effect.

SDCLKE Bit (SDCLK Enable)

This bit enables or disables output of the SDCLK pin.
Setting of the SDCLK bit should be changed, after the SDCLK setting has been set to the clock stop state.

21.2.26 Ethernet Control Register (PFENET)

Address(es): 0008 C10Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PHYMODE	—	—	—	—
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 always read as 0. The write value should always be 0.	R/W
b4	PHYMODE	Ethernet Mode Set	0: RMII mode 1: MII mode	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PHYMODE Bit (Ethernet Mode Set)

This bit specifies the PHY mode of the ETHERC.

Select the same mode as the one specified by the pin function select bits (PmnPFS.PSEL[4:0]).

When the signals for the RMII mode have been specified by the PmnPFS.PSEL[4:0] bits, set the PHYMODE bit to 0 (RMII mode), whereas when the signals for the MII mode have been specified by the PmnPFS.PSEL[4:0] bits, set the PHYMODE bit to 1 (MII mode).

21.2.27 USB0 Control Register (PFUSB0)

Address(es): 0008 C114h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	PUPHZS	PUPHZ Select	0: USB0_DPUPE pin: high-level output or low-level output (external pull-up control signal) 1: USB0_DPUPE pin: high-level output or Hi-Z state (USB0_DP pin pull-up output)	R/W
b3	PDHZS	PDHZ Select	0: USB0_DPRPD pin: high-level output or low-level output USB0_DRPD pin: high-level output or low-level output (external pull-down control signal) 1: USB0_DPRPD pin: low-level output or Hi-Z state USB0_DRPD pin: low-level output or Hi-Z state (USB0_DP/USB0_DM pin pull-down output)	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

When the DPUPE pin function has been assigned to a pin by the settings of the port mode register (PMR) and Pmn pin function control register (PmnPFS), the functions of the PUPHZS and PDHZS bits becomes effective.

PUPHZS Bit (PUPHZ Select)

This bit selects the output mode (external pull-up control/pin pull-up output) for the DPUPE pin of the USB0.
When the PUPHZS bit is set to 0, the control signal output mode for the external pull-up IC is selected and the high-active control signal is output from the DPUPE pin. When the DP pin of the USB0 is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the low-level output state.
When the PUPHZS bit is set to 1, the output mode where the DP pin is directly pulled-up is selected. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the Hi-Z state.

PDHZS Bit (PDHZ Select)

This bit selects the output mode (external pull-down control/pin pull-down output) for the DPRPD and DRPD pins of the USB0.
When the PDHZS bit is set to 0, the control signal output mode for the external pull-down IC is selected and the high-active control signal is output from the DPRPD and DRPD pins. When the DP and DM pins of the USB0 are pulled down, the DPRPD and DRPD pins are set to the high-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the low-level output state.
When the PDHZS bit is set to 1, the output mode where the DP and DM pins are directly pulled-down is selected. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the low-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the Hi-Z state.

21.2.28 USB1 Control Register (PFUSB1)

Address(es): 0008 C115h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	PUPHZS	PUPHZ Select	0: USB1_DPUPE pin: high-level output or low-level output (external pull-up control signal) 1: USB1_DPUPE pin: high-level output or Hi-Z state (USB1_DP pin pull-up output)	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PFUSB1 specifies the USB1 related input/output pins.
When the DPUPE pin function has been assigned to a pin by the settings of the Port mode register (PMR) and Pmn pin function control register (PmnPFS), the function of the PUPHZS bit becomes effective.

PUPHZS Bit (PUPHZ Select)

This bit selects the output mode (external pull-up control/pin pull-up output) for the DPUPE pin of the USB1.
When the PUPHZS bit is set to 0, the control signal output mode for the external pull-up IC is selected and the high-active control signal is output from the DPUPE pin. When the DP pin of the USB1 is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the low-level output state.
When the PUPHZS bit is set to 1, the output mode where the DP pin is directly pulled-up is selected. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the Hi-Z state.

21.3 How to Set the External Bus Interface

If the external bus interface is to be used, set the external bus enable bit (EXBE) in system control register 0 (SYSCR0) to 1 and then set the MPC registers.

Table 21.27 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 21.2, Register Descriptions.

Table 21.27 How to Set the External Bus Interface (1/3)

Port	Output Signal	Settings of MPC Registers
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	PFBCR1.ALEOE=1
P55	WAIT#	PFBCR1.WAITS[1:0]=01
P56	WR2# / BC2#	PFBCR0.WR32BC32E=1
P57	WR3# / BC3#	PFBCR0.WR32BC32E=1
	WAIT#	PFBCR1.WAITS[1:0]=00
P60	CS0#	PFCSE.CS0E=1, PFCSS0.CS0S=0
P61	CS1#	PFCSE.CS1E=1, PFCSS0.CS1S[1:0]=00
	SDCS#	PFBCR1.MDSDE=1
P62	CS2#	PFCSE.CS2E=1, PFCSS0.CS2S[1:0]=00
	RAS#	PFBCR1.MDSDE=1
P63	CS3#	PFCSE.CS3E=1, PFCSS0.CS3S[1:0]=00
	CAS#	PFBCR1.MDSDE=1
P64	CS4#	PFCSE.CS4E=1, PFCSS1.CS4S[1:0]=00
	WE#	PFBCR1.MDSDE=1
P65	CS5#	PFCSE.CS5E=1, PFCSS1.CS5S[1:0]=00
	CKE	PFBCR1.MDSDE=1
P66	CS6#	PFCSE.CS6E=1, PFCSS1.CS6S[1:0]=00
	DQM0	PFBCR1.MDSDE=1
P67	CS7#	PFCSE.CS7E=1, PFCSS1.CS7S[1:0]=00
	DQM1	PFBCR1.MDSDE=1
P70	SDCLK	PFBCR1.SDCLKE=1
P71	CS1#	PFCSE.CS1E=1, PFCSS0.CS1S[1:0]=01
P72	CS2#	PFCSE.CS2E=1, PFCSS0.CS2S[1:0]=01
P73	CS3#	PFCSE.CS3E=1, PFCSS0.CS3S[1:0]=01
P74	CS4#	PFCSE.CS4E=1, PFCSS1.CS4S[1:0]=01
P75	CS5#	PFCSE.CS5E=1, PFCSS1.CS5S[1:0]=01
P76	CS6#	PFCSE.CS6E=1, PFCSS1.CS6S[1:0]=01
P77	CS7#	PFCSE.CS7E=1, PFCSS1.CS7S[1:0]=01
P90	D16	PFBCR0.DH32E=1
	A16	PFAOE1.A16E=1, PFBCR0.ADRHMS=1

Table 21.27 How to Set the External Bus Interface (2/3)

Port	Output Signal	Settings of MPC Registers
P91	D17	PFBCR0.DH32E=1
	A17	PFAOE1.A17E=1, PFBCR0.ADRHMS=1
P92	D18	PFBCR0.DH32E=1
	A18	PFAOE1.A18E=1, PFBCR0.ADRHMS=1
P93	D19	PFBCR0.DH32E=1
	A19	PFAOE1.A19E=1, PFBCR0.ADRHMS=1
P94	D20	PFBCR0.DH32E=1
	A20	PFAOE1.A20E=1, PFBCR0.ADRHMS=1
P95	D21	PFBCR0.DH32E=1
	A21	PFAOE1.A21E=1, PFBCR0.ADRHMS=1
P96	D22	PFBCR0.DH32E = 1
	A22	PFAOE1.A22E=1, PFBCR0.ADRHMS=1
P97	D23	PFBCR0.DH32E=1
	A23	PFAOE1.A23E=1, PFBCR0.ADRHMS=1
PA0	A0	PFBCR0.ADRLE=1
	BC0#	PFBCR0.ADRLE=1, CSnMOD.WRMOD=1, SDCCR.EXENB=1, SDCCR.BSIZE[1:0]=00/10/11 or PFBCR0.ADRLE=1, CSnMOD.WRMOD=1, SDCCR.EXENB=0
	DQM2	PFBCR0.ADRLE=1, SDCCR.EXENB=1, SDCCR.BSIZE[1:0]=01
PA1	A1	PFBCR0.ADRLE=1
	DQM3	PFBCR0.ADRLE=1, SDCCR.EXENB=1, SDCCR.BSIZE[1:0]=01
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
PC2	A18	PFAOE1.A18E=1, PFBCR0.ADRHMS=0
PC3	A19	PFAOE1.A19E=1, PFBCR0.ADRHMS=0
PC4	A20	PFAOE1.A20E=1, PFBCR0.ADRHMS=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

Table 21.27 How to Set the External Bus Interface (3/3)

Port	Output Signal	Settings of MPC Registers
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
PG1	D25	PFBCR0.DH32E=1
PG2	D26	PFBCR0.DH32E=1
PG3	D27	PFBCR0.DH32E=1
PG4	D28	PFBCR0.DH32E=1
PG5	D29	PFBCR0.DH32E=1
PG6	D30	PFBCR0.DH32E=1
PG7	D31	PFBCR0.DH32E=1

21.4 Usage Notes

21.4.1 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 input function.
(2) Enable writing to the Pmn pin function control register (PmnPFS) through the write-protect register (PWPR) setting. (m = 0 to 9, A to F, and J, n = 0 to 7)
(3) Specify the input/output function for the pin through the PSEL[4:0] bit settings in the PFS register.
(4) Clear the PFSWE bit in the PWPR register to 0 to disable writing to the PmnPFS register.
(5) Set the PMR to 1 as necessary to switch to the selected input/output function for the pin.

21.4.2 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 cleared 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 pins for the A/D converter and D/A converter. When using these ports as analog input 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 21.28. 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[4:0] bits are made.

Table 21.28 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[4:0]	
After a reset	0	0	0	0	00000b	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	0/1	x	Set the PmnPFS.ISEL bit to 1 if these are multiplexed with interrupt inputs.
Peripheral functions	1	x	0	0/1	Peripheral functions (see Table 21.1 to Table 21.26)	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*	x	Register settings are not required.
Analog inputs and outputs	0	0	1	x*	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	0	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
JTAG-IF	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*	x	Set these as general input port pins so that the output buffers are turned off.

Table 21.28 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[4:0]	
XCIN/XCOUT	0	0	x	x*	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: • Even if the PmnPFS.ISEL bit is set to 1, the pin will not function as an IRQn input pin.

Note 1. The pin state is readable when the PmnPFS.ASEL bit is 0.
Note 2. If the value of the PmnPFS.PSEL[4:0] bits is to be changed, do so while the PMR.Bn bit is 0.
Note 3. 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.
Note 4. 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) to 0 (disabled).
Note 5. The order of priority for output functions is as follows.
Debugger (JTAG/FINE/tracing) > external bus > general port > peripheral modules
Note 6. If the following functions are simultaneously set up for an external bus pin, the order of priority is as follows.
Data bus (1) > address bus, ADRAM control signal (2) > CS# (3)
(1) D0 to D31 (data bus)
(2) RD#, WR0#/WR# to WR3#, BC0# to BC3#, BCLK, SDCLK, SDCS#, RAS#, CAS#, WE#, CKE, DQM0 to DQM3, A0 to A23 (address bus)
(3) CS0# to CS7# (chip select)

21.4.3 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[1:0]) to 1.

22. Multi-Function Timer Pulse Unit 2 (MTU2a)

22.1 Overview

The RX63N/RX631 Group has an on-chip multi-function timer pulse unit 2 (MTU). Each unit comprises a 16-bit timer with six channels (MTU0 to MTU5).
Table 22.1 lists the specifications of the MTU, and Table 22.2 lists the function list. Figure 22.1 shows a block diagram of the MTU.

Table 22.1 Specifications of MTU

Item	Description
Pulse input/output	16 lines max.
Pulse input	3 lines
Count clock	Eight clocks or seven clocks for each channel (four clocks for MTU5)
Available operations	[MTU0 to MTU4] • Waveform output at compare match • Input capture function (noise filter set function) • Counter clear operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing by compare match or input capture • Simultaneous register input/output by synchronous counter operation • A maximum of 12-phase PWM output is available in combination with synchronous operation
	[MTU0, MTU3, MTU4] • Buffer operation specifiable • AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset-synchronized PWM output is settable and the selection of two types of waveform outputs (chopping and level) is possible.
	[MTU1, MTU2] • Phase counting mode specifiable independently • Cascade connection operation
	[MTU3, MTU4] • A total of six-phase waveform output, which includes three phases each for positive and negative complementary PWM or reset PWM output, by interlocking operation
	[MTU5] • Dead time compensation counter
Complementary PWM mode	• Interrupts at the crest and trough of the counter value • A/D converter start triggers can be skipped
Interrupt sources	28 sources
Buffer operation	Automatic transfer of register data
Trigger generation	Programmable pulse generator (PPG) output trigger can be generated
	A/D converter start trigger can be generated
Power reduction	Each unit can be independently set to module stop stat

Table 22.2 MTU Functions (1/2)

Item		MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Count clock		PCLK/1 PCLK/4 PCLK/16 PCLK/64 MTCLKA MTCLKB MTCLKC MTCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 MTCLKA MTCLKB MTCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW
General registers/ buffer registers		TGRC TGRD TGRF	—	—	TGRC TGRD	TGRC TGRD	—
I/O pins		MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	Input pins MTIC5U MTIC5V MTIC5W
Counter clear function		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	○	○	○	○	○	—
	1 output	○	○	○	○	○	—
	Toggle output	○	○	○	○	○	—
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	—
PWM mode 1		○	○	○	○	○	—
PWM mode 2		○	○	○	—	—	—
Complementary PWM mode		—	—	—	○	○	—
Reset-synchronized PWM		—	—	—	○	○	—
AC synchronous motor drive mode		○	—	—	○	○	—
Phase counting mode		—	○	○	—	—	—
Buffer operation		○	—	—	○	○	—
Dead time compensation counter function		—	—	—	—	—	○
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	—
DTC activation		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 and TCNT overflow or underflow	TGR compare match or input capture

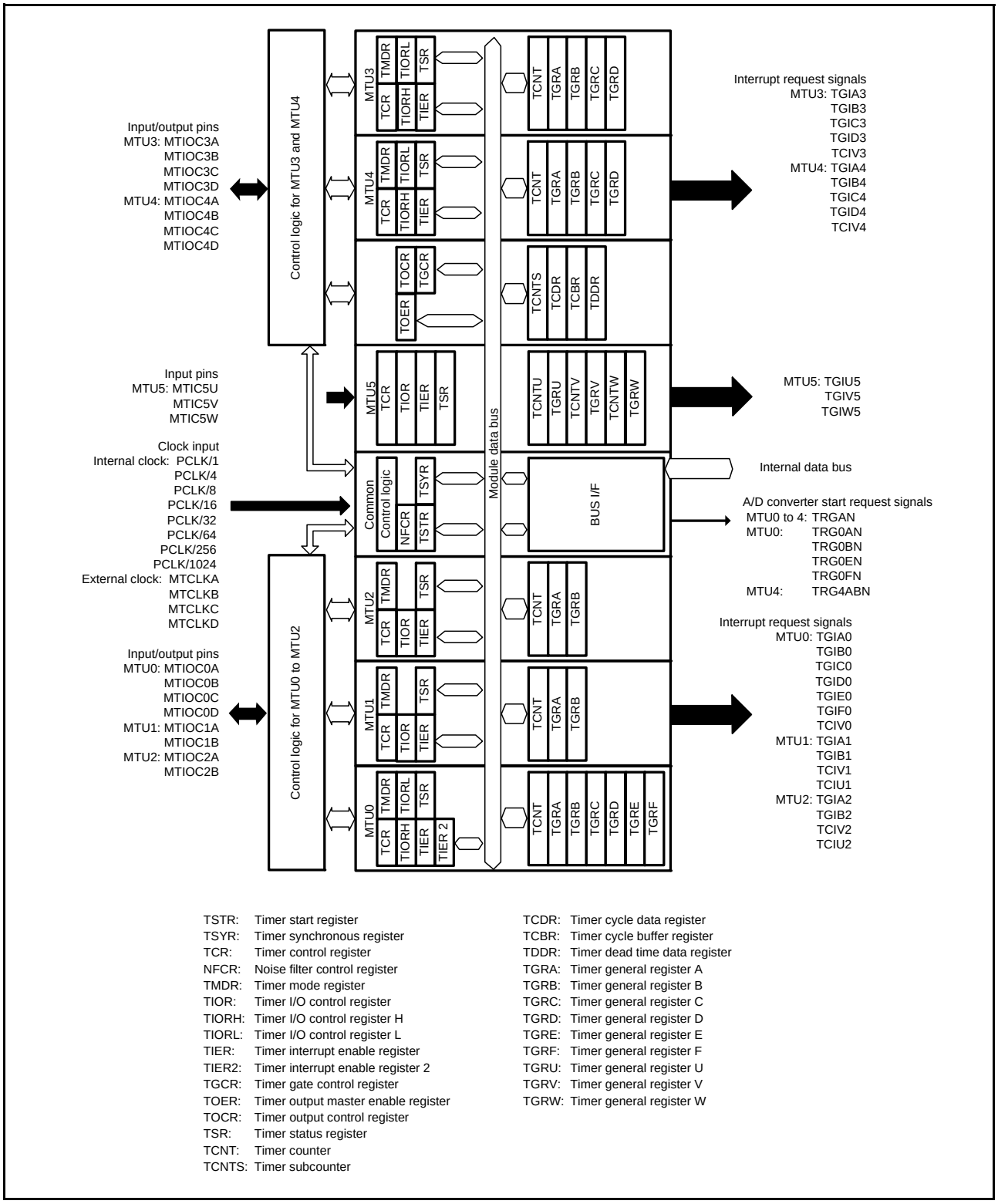
Table 22.2 MTU Functions (2/2)

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
A/D converter start trigger	TGRA compare match or input capture TGRB compare match or input capture TGRC compare match or input capture TGRD compare match or input capture TGRE compare match TGRF compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture TCNT underflow (trough) in complementary PWM mode	—
PPG trigger	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	—	—
Interrupt sources	7 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 match 0E • Compare match 0F • 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	5 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	3 sources • Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W
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	—
Interrupt skipping function	—	—	—	• Skips TGRA compare match interrupts	• Skips TCIV interrupts	—
Module stop function	MSTPCRA.MSTPA9*1					

○: Possible

—: Not possible

Note 1. For details on the module stop function, see section 11, Low Power Consumption.



TSTR: Timer start register
TSYR: Timer synchronous register
TCR: Timer control register
NFCR: Noise filter control register
TMDR: Timer mode register
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
TGCR: Timer gate control register
TOER: Timer output master enable register
TOCR: Timer output control register
TSR: Timer status register
TCNT: Timer counter
TCNTS: Timer subcounter
TCDR: Timer cycle data register
TCBR: Timer cycle buffer register
TDDR: Timer dead time data register
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

Figure 22.1 Block Diagram of MTU

Table 22.3 lists the pin configuration of the MTU.

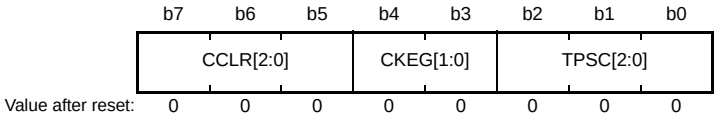
Table 22.3 Pin Configuration of MTU

Module Symbol	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 (MTU2 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)
MTU0	MTIOC0A	I/O	TGRA0 input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	TGRB0 input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	TGRC0 input capture input/output compare output/PWM output pin
	MTIOC0D	I/O	TGRD0 input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	TGRA1 input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	TGRB1 input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	TGRA2 input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	TGRB2 input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	TGRA3 input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	TGRB3 input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	TGRC3 input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	TGRD3 input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	TGRA4 input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	TGRB4 input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	TGRC4 input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	TGRD4 input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	TGRU5 input capture input/external pulse input pin
	MTIC5V	Input	TGRV5 input capture input/external pulse input pin
	MTIC5W	Input	TGRW5 input capture input/external pulse input pin

22.2 Register Descriptions

22.2.1 Timer Control Register (TCR)

Address(es): MTU0.TCR 0008 8700h, MTU1.TCR 0008 8780h, MTU2.TCR 0008 8800h, MTU3.TCR 0008 8600h, MTU4.TCR 0008 8601h, MTU5.TCRU 0008 8884h, MTU5.TCRV 0008 8894h, MTU5.TCRW 0008 88A4h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	See Table 22.6 to Table 22.10.	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	See Table 22.4 and Table 22.5.	R/W

x: Don't care

The MTU has a total of eight TCR registers, one each for MTU0 to MTU4 and three (TCRU, TCRV, and TCRW) for MTU5.
TCR is an 8-bit readable/writable register that controls the TCNT operation for each channel. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See Table 22.6 to Table 22.10 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the input clock edge. When the input clock is counted at both edges, the input clock period is halved (e.g. PCLK/4 at both edges = PCLK/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 input clock is PCLK/4 or slower. When PCLK/1 or the overflow/underflow in another channel is selected for the input clock, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] Bits (Counter Clear)

These bits select the TCNT counter clearing source. See Table 22.4 and Table 22.5 for details.

Table 22.4 CCLR[2:0] (MTU0, MTU3, and MTU4)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
MTU0, MTU3, MTU4	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 TSYR.SYNC bit to 1.
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 22.5 CCLR[2:0] (MTU1 and MTU2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR1	Bit 5 CCLR0	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 TSYR.SYNC bit to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. It is always read as 0. The write value should be 0.

Table 22.6 TTPSC[2:0] (MTU0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU0	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 MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	External clock: counts on MTCLKC pin input
	1	1	1	External clock: counts on MTCLKD pin input

Table 22.7 TPSC[2:0] (MTU1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU1	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 MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Counts on MTU2.TCNT overflow/underflow

Note: • This setting is ignored when MTU1 is in phase counting mode.

Table 22.8 TPSC[2:0] (MTU2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU2	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 MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	External clock: counts on MTCLKC pin input
	1	1	1	Internal clock: counts on PCLK/1024

Note: • This setting is ignored when MTU2 is in phase counting mode.

Table 22.9 TPSC[2:0] (MTU3 and MTU4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU3, MTU4	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	Internal clock: counts on PCLK/256
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	External clock: counts on MTCLKA pin input
	1	1	1	External clock: counts on MTCLKB pin input

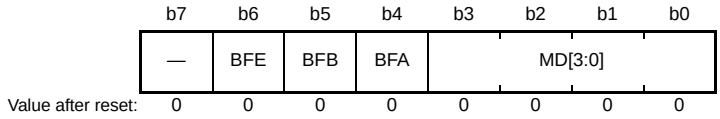
Table 22.10 TPSC[1:0] (MTU5)

Channel	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU5	0	0	Internal clock: counts on PCLK/1
	0	1	Internal clock: counts on PCLK/4
	1	0	Internal clock: counts on PCLK/16
	1	1	Internal clock: counts on PCLK/64

Note: • Bits 7 to 2 are reserved in MTU5. These bits are always read as 0. The write value should be 0.

22.2.2 Timer Mode Register (TMDR)

Address(es): MTU0.TMDR 0008 8701h, MTU1.TMDR 0008 8781h, MTU2.TMDR 0008 8801h, MTU3.TMDR 0008 8602h, MTU4.TMDR 0008 8603h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. See Table 22.11 for details.	R/W
b4	BFA	Buffer Operation A	0: TGRA and TGRC operate in normal mode 1: TGRA and TGRC used together for buffer operation	R/W
b5	BFB	Buffer Operation B	0: TGRB and TGRD operate in normal mode 1: TGRB and TGRD used together for buffer operation	R/W
b6	BFE	Buffer Operation E	0: MTU0.TGRE and MTU0.TGRF operate in normal mode 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

TMDR is an 8-bit readable/writable register that specifies the operating mode of each channel. TMDR values should be specified only while TCNT operation is stopped.

Table 22.11 Operating Mode Setting by MD[3:0] Bits

Bit 3	Bit 2	Bit 1	Bit 0	
MD3	**MD2**	**MD1**	**MD0**	**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*1
0	1	0	0	Phase counting mode 1*2
0	1	0	1	Phase counting mode 2*2
0	1	1	0	Phase counting mode 3*2
0	1	1	1	Phase counting mode 4*2
1	0	0	0	Reset-synchronized PWM mode*3
1	0	0	1	Setting prohibited
1	0	1	x	Setting prohibited
1	1	0	0	Setting prohibited
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*3
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*3
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*3

x: Don't care

Note 1. PWM mode 2 cannot be set for MTU3 and MTU4.

Note 2. Phase counting mode cannot be set for MTU0, MTU3, and MTU4.

Note 3. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3.
When MTU3 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 settings become ineffective and automatically conform to the MTU3 setting, respectively. The initial values should be set for MTU4.
Reset-synchronized PWM mode and complementary PWM mode cannot be set for MTU0, MTU1 and MTU2.

BFA Bit (Buffer Operation A)

This bit specifies whether to operate TGRA in normal mode 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 cleared to 0.

When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the BFA bit in MTU4.TMDR 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 22.40 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This bit specifies whether to operate TGRB in normal mode 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 in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register 3 or 4 (MTU3.TIER or MTU4.TIER) should be cleared to 0.

When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the BFB bit in MTU4.TMDR 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 22.40 for an illustration of the Tb interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

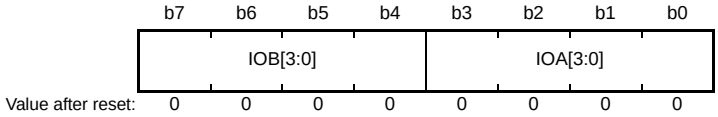
This bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in normal mode or to use them together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register.

In MTU1 to MTU4, this bit is reserved. It is read as 0. The write value should be 0.

22.2.3 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH

Address(es): MTU0.TIORH 0008 8702h, MTU1.TIOR 0008 8782h, MTU2.TIOR 0008 8802h, MTU3.TIORH 0008 8604h, MTU4.TIORH 0008 8606h

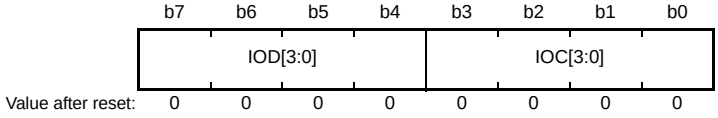


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A	See the following tables.*1 MTU0.TIORH: Table 22.20 MTU1.TIOR: Table 22.22 MTU2.TIOR: Table 22.23 MTU3.TIORH: Table 22.24 MTU4.TIORH: Table 22.26	R/W
b7 to b4	IOB[3:0]	I/O Control B	See the following tables.*1 MTU0.TIORH: Table 22.12 MTU1.TIOR: Table 22.14 MTU2.TIOR: Table 22.15 MTU3.TIORH: Table 22.16 MTU4.TIORH: Table 22.18	R/W

Note 1. If the IOn[3:0] (n = A, B) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level or toggling of the output in response to compare matches is in progress, the output becomes high impedance.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL

Address(es): MTU0.TIORL 0008 8703h, MTU3.TIORL 0008 8605h, MTU4.TIORL 0008 8607h

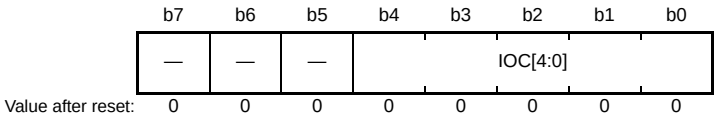


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C	See the following tables.*1 MTU0.TIORL: Table 22.21 MTU3.TIORL: Table 22.25 MTU4.TIORL: Table 22.27	R/W
b7 to b4	IOD[3:0]	I/O Control D	See the following tables.*1 MTU0.TIORL: Table 22.13 MTU3.TIORL: Table 22.17 MTU4.TIORL: Table 22.19	R/W

Note 1. If the IOn[3:0] (n = C, D) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level or toggling of the output in response to compare matches is in progress, the output becomes high impedance.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 0008 8886h, MTU5.TIORV 0008 8896h, MTU5.TIORW 0008 88A6h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IOC[4:0]	I/O Control C	See the following table. MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 22.28	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MTU has a total of 11 TIOR registers, two each for MTU0, MTU3, and MTU4, one each for MTU1 and MTU2, and three (MTU5.TIORU/V/W) each for MTU5.
TIOR should be set when TMDR is set to select normal mode, PWM mode, or phase counting mode.
The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is 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 22.12 TIORH (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU0.TGRB Function**	**MTIOC0B Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Table 22.13 TIORL (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	**IOD2**	**IOD1**	**IOD0**	**MTU0.TGRD Function**	**MTIOC0D Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Note 1. When the MTU0.TMDR.BFB is set to 1 and MTU0.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.14 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU1.TGRB Function**	**MTIOC1B Pin Function**
0	0	0	0	MTU1.TGRB works as an output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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.TGRC compare match/input capture.

x: Don't care

Table 22.15 TIOR (MTU2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU2.TGRB Function**	**MTIOC2B Pin Function**
0	0	0	0	MTU2.TGRB works as an output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.16 TIORH (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU3.TGRB Function**	**MTIOC3B Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.17 TIORL (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	**IOD2**	**IOD1**	**IOD0**	**MTU3.TGRD Function**	**MTIOC3D Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 BFB bit in MTU3.TMDR is set to 1 and MTU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.18 TIORH (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU4.TGRB Function**	**MTIOC4B Pin Function**
0	0	0	0	MTU4.TGRB works as an output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.19 TIORL (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	**IOD2**	**IOD1**	**IOD0**	**MTU4.TGRD Function**	**MTIOC4D Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 BFB bit in MTU4.TMDR is set to 1 and MTU4.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.20 TIORH (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU0.TGRA Function**	**MTIOC0A Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Table 22.21 TIORL (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	**IOC2**	**IOC1**	**IOC0**	**MTU0.TGRC Function**	**MTIOC0C Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Note 1. When the BFA bit in MTU0.TMDR is set to 1 and MTU0.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.22 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU1.TGRA Function**	**MTIOC1A Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.23 TIOR (MTU2)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU2.TGRA Function**	**MTIOC2A Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.24 TIORH (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU3.TGRA Function**	**MTIOC3A Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.25 TIORL (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	**IOC2**	**IOC1**	**IOC0**	**MTU3.TGRC Function**	**MTIOC3C Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 BFA bit in MTU3.TMDR is set to 1 and MTU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.26 TIORH (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU4.TGRA Function**	**MTIOC4A Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.27 TIORL (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	**IOC2**	**IOC1**	**IOC0**	**MTU4.TGRC Function**	**MTIOC4C Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 BFA bit in MTU4.TMDR is set to 1 and MTU4.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.28 TIORU, TIORV, and TIORW (MTU5)

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC4	IOC3	IOC2	IOC1	IOC0	MTU5.TGRU, MTU5.TGRV, MTU5.TGRW Function	MTIC5U, MTIC5V, MTIC5W Pin Function
0	0	0	0	0	Compare match register	Compare match
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	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		Setting prohibited
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

22.2.4 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 0008 88B6h

	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 is an 8-bit readable/writable register that specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW.

22.2.5 Timer Interrupt Enable Register (TIER)

- TIER (MTU0 to MTU4)

Address(es): MTU0.TIER 0008 8704h, MTU1.TIER 0008 8784h, MTU2.TIER 0008 8804h, MTU3.TIER 0008 8608h, MTU4.TIER 0008 8609h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	TTGE2	TCIEU	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 MTU4.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTU4.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

The MTU has a total of seven TIER registers, two each for MTU0 and one each for MTU1 to MTU5.
TIER is an 8-bit readable/writable register that enables or disables interrupt requests in each channel.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A or B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables interrupt requests (TGIn) in MTU0, MTU3 and MTU4 (n = C or 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 MTU1 and MTU2.
In MTU0, MTU3, and MTU4, 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 MTU4.TCNT underflow (trough) in complementary PWM mode.
In MTU0 to MTU3, 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.

- TIER2 (MTU0)

Address(es): MTU0.TEIR2 0008 8724h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	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
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	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.TGRm (m = E or F).

- TIER (MTU5)

Address(es): MTU5.TIER 0000 88B2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIE5W	TGR Interrupt Enable 5W	0: Interrupt requests TGI5W disabled 1: Interrupt requests TGI5W enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGI5V disabled 1: Interrupt requests TGI5V enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGI5U disabled 1: Interrupt requests TGI5U enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TGIE5W, TGIE5V, and TGIE5U Bits (TGR Interrupt Enable 5m)

Each bit enables or disables interrupt requests (TGIm5) (m = W, V, or U).

22.2.6 Timer Status Register (TSR)

- TSR (MTU0 to MTU4)

Address(es): MTU0.TSR 0008 8705h, MTU1.TSR 0008 8785h, MTU2.TSR 0008 8805h, MTU3.TSR 0008 862Ch, MTU4.TSR 0008 862Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 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

The MTU has a total of five TSR registers, one each for MTU0 to MTU4.
TSR is an 8-bit readable/writable register that indicates the status of each channel.

TCFD Flag (Count Direction Flag)

Status flag that shows the direction in which TCNT counts in MTU1 to MTU4.
In MTU0, this bit is reserved. It is read as 1. The write value should be 1.

22.2.7 Timer Buffer Operation Transfer Mode Register (TBTM)

Address(es): MTU0.TBTM 0008 8726h, MTU3.TBTM 0008 8638h, MTU4.TBTM 0008 8639h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	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 in MTU0, 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

The MTU has a total of three TBTM registers, one each for MTU0, MTU3 and MTU4.
TBTM is an 8-bit readable/writable register that specifies the timing for transferring data from the buffer register to the timer general register in PWM mode.

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 and MTU4, this bit is reserved. It is read as 0 and the write value should be 0. When MTU0 is not set to PWM mode, do not set the TTSE bit to 1.

22.2.8 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 0008 8790h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

x: Undefined

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 TMTIOC1B 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

The MTU has one TICCR for MTU1.

MTICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded.

22.2.9 Timer A/D Converter Start Request Control Register (TADCR)

Address(es): MTU4.TADCR 0008 8640h

	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*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1

Note 1. Do not set to 1 when complementary PWM mode is not selected.

Bit	Symbol	Bit Name	Description	R/W
b0	ITB4VE	TCIV4 Interrupt Skipping Link Enable	0: TCI4V interrupt skipping is not linked 1: TCI4V interrupt skipping is linked	R/W
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable	0: TCI4V interrupt skipping is not linked 1: TCI4V interrupt skipping is linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W
b4	DT4BE	Down-Count TRG4BN Enable	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	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 start requests (TRG4AN) enabled during MTU4.TCNT up-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/B Transfer Timing Select	See Table 22.29 for details.	R/W

Note 1. TADCR must not be accessed in 8-bit units; it should be accessed in 16-bit units.
Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
Note 3. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 4. Do not initialize b6 to b0 to 1 when complementary PWM mode is not selected.

TADCR is a 16-bit readable/writable register that enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation.

Table 22.29 Setting of Transfer Timing by BF0[1:0] Bits

Bit 7	Bit 6	
BF1	**BF0**	**Description**
0	0	Does not transfer data from the cycle set buffer register to the cycle set register.
0	1	Transfers data from the cycle set buffer register to the cycle set register at the crest of the MTU4.TCNT count.*1
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTU4.TCNT count.*2
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the MTU4.TCNT count.*2

Note 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the MTU4.TCNT count is reached in complementary PWM mode, when a compare match occurs between MTU3.TCNT and MTU3.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTU4.TCNT and MTU4.TGRA in PWM mode 1 or normal mode.
Note 2. These settings are prohibited when complementary PWM mode is not selected.

22.2.10 Timer A/D Converter Start Request Cycle Set Registers A and B (TADCORA and TADCORB)

Address(es): MTU4.TADCORA 0008 8644h, MTU4.TADCORB 0008 8646h

	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 1. TADCORA_4 and TADCORB_4 must not be accessed in 8-bit units; they should always be accessed in 16-bit units.

TADCORA and TADCORB are 16-bit readable/writable registers. When the MTU4.TCNT count reaches the value in TADCORA or TADCORB, a corresponding A/D converter start request will be issued.
The TADCORA and TADCORB values after reset are FFFFh.

22.2.11 Timer A/D Converter Start Request Cycle Set Buffer Registers A and B (TADCOBRA and TADCOBRB)

Address(es): MTU4.TADCOBRA 0008 8648h, MTU4.TADCOBRB 0008 864Ah

	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 1. TADCORA_4 and TADCORB_4 must not be accessed in 8-bit units; they should always be accessed in 16-bit units.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers. When the crest or trough of the MTU4.TCNT count is reached, these register values are transferred to TADCORA and TADCORB, respectively.
The TADCOBRA and TADCOBRB values after reset are FFFFh.

22.2.12 Timer Counter (TCNT)

Address(es): MTU0.TCNT 0008 8706h, MTU1.TCNT 0008 8786h, MTU2.TCNT 0008 8806h, MTU3.TCNT 0008 8610h, MTU4.TCNT 0008 8612h, MTU5.TCNTU 0008 8880h, MTU5.TCNTV 0008 8890h, MTU5.TCNTW 0008 88A0h

	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 1. The TCNT counters must not be accessed in 8-bit units; they should always be accessed in 16-bit units.

The MTU has a total of eight TCNT counters, one each for MTU0 to MTU4 and three (MTU5.TCNTU, TCNTV, and TCNTW) for MTU5. TCNT is a 16-bit readable/writable counter.
TCNT is initialized to 0000h by a reset.

22.2.13 Timer General Register (TGR)

Address(es): MTU0.TGRA 0008 8708h, MTU0.TGRB 0008 870Ah, MTU0.TGRC 0008 870Ch, MTU0.TGRD 0008 870Eh, MTU0.TGRE 0008 8720h, MTU0.TGRF 0008 8722h, MTU1.TGRA 0008 8788h, MTU1.TGRB 0008 878Ah, MTU2.TGRA 0008 8808h, MTU2.TGRB 0008 880Ah, MTU3.TGRA 0008 8618h, MTU3.TGRB 0008 861Ah, MTU3.TGRC 0008 8624h, MTU3.TGRD 0008 8626h, MTU4.TGRA 0008 861Ch, MTU4.TGRB 0008 861Eh, MTU4.TGRC 0008 8628h, MTU4.TGRD 0008 862Ah, MTU5.TGRU 0008 8882h, MTU5.TGRV 0008 8892h, MTU5.TGRW 0008 88A2h

	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 1. The TGR registers must not be accessed in 8-bit units; they should always be accessed in 16-bit units. TGR registers are initialized to FFFFh.

The MTU has a total of 21 TGR registers, six for MTU0, two each for MTU1 and MTU2, four each for MTU3 and MTU4, and three for MTU5. TGR is a 16-bit readable/writable register.
TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for MTU0, MTU3, and MTU4 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. TGRF 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.

22.2.14 Timer Start Registers (TSTR)

- TSTR (MTU0 to MTU4)

Address(es): MTU.TSTR 0008 8680h

	b7	b6	b5	b4	b3	b2	b1	b0
	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: MTU0.TCNT performs count stop 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT performs count stop 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT performs count stop 1: MTU2.TCNT performs count operation	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST3	Counter Start 3	0: MTU3.TCNT performs count stop 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT performs count stop 1: MTU4.TCNT performs count operation	R/W

TSTR starts or stops TCNT operation in MTU0 to MTU4.
Before setting the operating mode in TMDR or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

CSTn Bits (Counter Start n) (n = 0 to 4)

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.

- TSTR (MTU5)

Address(es): MTU5.TSTR 0008 88B4h

	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 count operation is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV count operation is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU count operation 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

22.2.15 Timer Synchronous Registers (TSYR)

Address(es): MTU.TSYR 0008 8681h

	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 presetting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. TCNT synchronous presetting/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 presetting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W

TSYR selects independent operation or synchronous operation of TCNT in MTU0 to MTU4.
A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

SYNCn Bits (Timer Synchronous n Operation) (n = 0 to 4)

Each bit selects whether operation is independent of or synchronized with other channels.
When synchronous operation is selected, the TCNT synchronous presetting of multiple channels and synchronous clearing by counter clearing on another channel are possible.
To set synchronous operation, the SYNCn bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNCn bit, the TCNT clearing source must also be set by means of TCR.CCLR[2:0] bits.

22.2.16 Timer Read/Write Enable Registers (TRWER)

Address(es): MTU.TRWER 0008 8684h

	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

TRWER enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.
[Clearing conditions]

- When 0 is written to the RWE bit after reading RWE bit = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification

22 registers: MTUm.TCR, MTUm.TMDR, MTUm.TIORH, MTUm.TIORL, MTUm.TIER, MTUm.TGRA, MTUm.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, and MTUm.TCNT (m = 3, 4)

22.2.17 Timer Output Master Enable Registers (TOER)

Address(es): MTU.TOER 0008 860Ah

	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 (inactive level)*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b7 to b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The inactive level is determined by the settings in timer output control registers 1 and 2 (TOCR1 and TOCR2). For details, refer to section 22.2.18, Timer Output Control Registers 1 (TOCR1), and section 22.2.19, Timer Output Control Registers 2 (TOCR2). Set these bits to 1 to enable MTU output when complementary PWM or reset-synchronized PWM mode is not selected. When these bits are set to 0, the inactive level is output by setting the timer output control registers 1 and 2 (TOCR1 and TOCR2).

TOER enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the TOER bits have not been set. In MTU3 and MTU4, set TOER prior to setting TIOR.

22.2.18 Timer Output Control Registers 1 (TOCR1)

Address(es): MTU.TOCR 0008 860Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0*1	0	0	0

Note 1. This bit can be set to 1 only once after a power-on reset. After 1 is written, 0 cannot be written to the bit.

Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P*2	See Table 22.30.	R/W
b1	OLSN	Output Level Select N*2	See Table 22.31.	R/W
b2	TOCS	TOC Select	0: TOCR1 setting is selected 1: TOCR2 setting is selected	R/W
b3	TOCL	TOC Register Write Protection*1	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*3
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. Setting the TOCR1.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 2. Clearing the TOCR1.TOCS bit to 0 makes this bit setting valid.
Note 3. This bit can be set to 1 only once after a power on reset. After 1 is written, 0 cannot be written to the bit.

TOCR1 is 8-bit readable/writable registers that 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.

OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode.

TOCS Bit (TOC Select)

This bit selects either the TOCR1 or TOCR2 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 TOCR1.

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM cycle.

Table 22.30 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Outputl	Active Leve	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 22.31 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Outputl	Active Leve	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 22.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

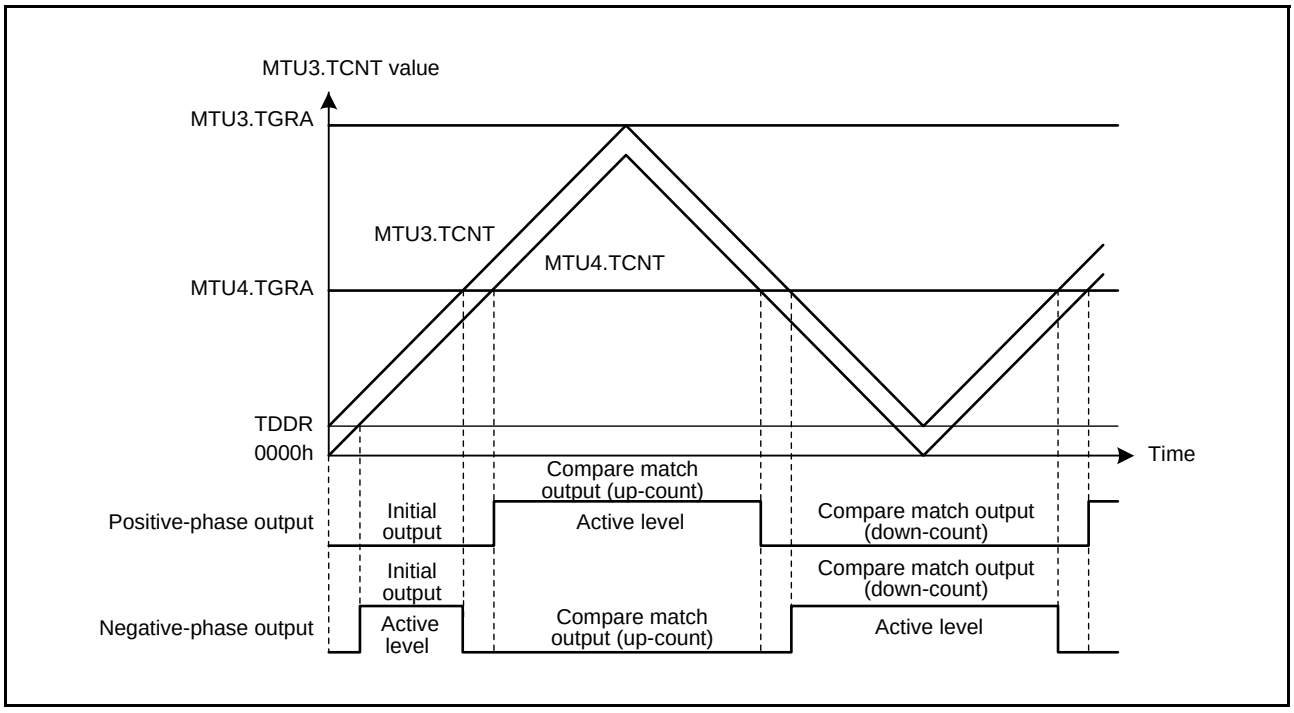


Figure 22.2 Example of Output in Complementary PWM Mode

22.2.19 Timer Output Control Registers 2 (TOCR2)

Address(es): MTU.TOCR2 0008 860Fh

	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	This bit selects the output level on MTIOC3B in reset-synchronized PWM mode and complementary PWM mode. See Table 22.32.	R/W
b1	OLS1N	Output Level Select 1N*1	This bit selects the output level on MTIOC3D in reset-synchronized PWM mode and complementary PWM mode. See Table 22.33.	R/W
b2	OLS2P	Output Level Select 2P*1	This bit selects the output level on MTIOC4A in reset-synchronized PWM mode and complementary PWM mode. See Table 22.34.	R/W
b3	OLS2N	Output Level Select 2N*1	This bit selects the output level on MTIOC4C in reset-synchronized PWM mode and complementary PWM mode. See Table 22.35.	R/W
b4	OLS3P	Output Level Select 3P*1	This bit selects the output level on MTIOC4B in reset-synchronized PWM mode and complementary PWM mode. See Table 22.36.	R/W
b5	OLS3N	Output Level Select 3N*1	This bit selects the output level on MTIOC4D in reset-synchronized PWM mode and complementary PWM mode. See Table 22.37.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBR to TOCR2. See Table 22.38 for details.	R/W

Note 1. Setting the TOCR1.TOCS bit to 1 makes this bit setting valid.

TOCR2 control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Table 22.32 MTIOC3B 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

Table 22.33 MTIOC3D 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

Note: • The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.34 MTIOC4A 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

Table 22.35 MTIOC4C 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

Note: • The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.36 MTIOC4B 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

Table 22.37 MTIOC4D 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

Note: • The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.38 Setting of TOCR2.BF[1:0] Bits

Bit 7	Bit 6	Description	
BF1	BF0	Complementary PWM Mode	Reset-Synchronized PWM Mode
0	0	Does not transfer data from the buffer register (TOLBR) to TOCR2.	Does not transfer data from the buffer register (TOLBR) to TOCR2.
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the MTU4.TCNT count.	Transfers data from the buffer register (TOLBR) to TOCR2 when MTU4.TCNT or MTU3.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBRj) to TOCR2 at the trough of the MTU4.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the MTU4.TCNT count.	Setting prohibited

22.2.20 Timer Output Level Buffer Registers (TOLBR)

Address(es): MTU.TOLBR 0008 8636h

	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 TOCR2.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TOLBR is 8-bit readable/writable registers that function as buffer registers for TOCR2 and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 22.3 shows an example of the PWM output level setting procedure in buffer operation.

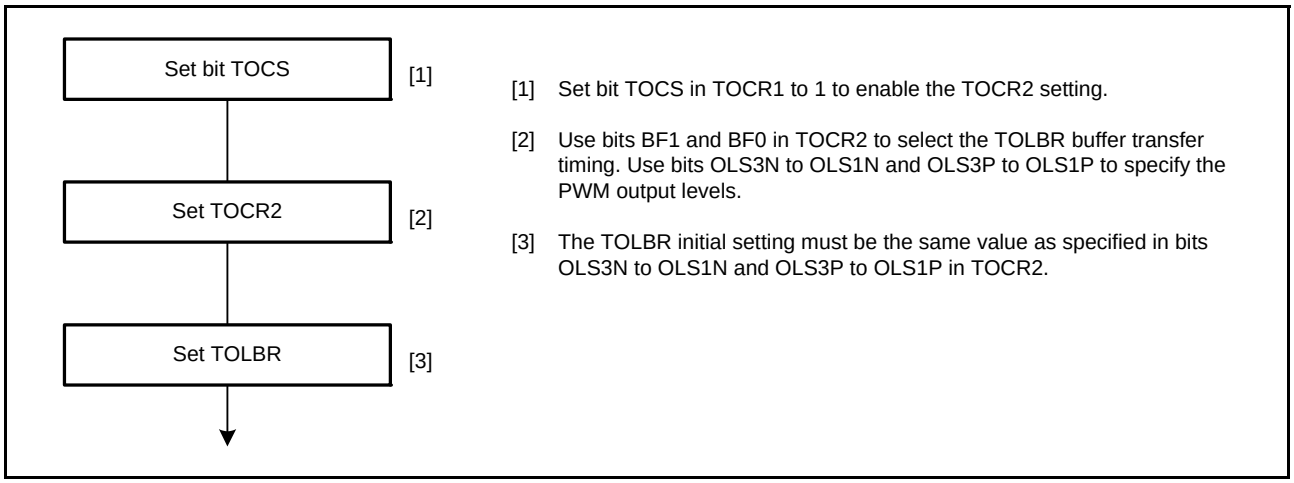


Figure 22.3 Example of PWM Output Level Setting Procedure in Buffer Operation

22.2.21 Timer Gate Control Registers (TGCR)

Address(es): MTU.TGCR 0008 860Dh

	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 TGCR.FB bit s set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 22.39.	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 (TGCR'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

TGCR control the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. These 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 TGCR.FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 22.39.

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 TGCR.

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 TGCR effective or ineffective.

Table 22.39 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

22.2.22 Timer Subcounters (TCNTS)

Address(es): MTU.TCNTS 0008 8620h

	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: • Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

TCNTS is 16-bit read-only counters that are used only in complementary PWM mode.
The TCNTS value after reset is 0000h.

22.2.23 Timer Dead Time Data Registers (TDDR)

Address(es): MTU.TDDR 0008 8616h

	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: • Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

TDDR is 16-bit registers, used only in complementary PWM mode, that specify the MTU3.TCNT and MTU4.TCNT counter offset value. In complementary PWM mode, when the MTU3.TCNT and MTU4.TCNT counters are cleared and then restarted, the TDDR value is loaded into the MTU3.TCNT counter and the count operation starts. The TDDR value after reset is FFFFh.

22.2.24 Timer Cycle Data Registers (TCDR)

Address(es): MTU.TCDR 0008 8614h

	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: • Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

TCDR is 16-bit registers used only in complementary PWM mode. Set half the PWM carrier cycle as the TCDR value. TCDR is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (down-count to up-count). The TCDR value after reset is FFFFh.

22.2.25 Timer Cycle Buffer Registers (TCBR)

Address(es): MTU.TCBR 0008 8622h

	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: • Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

TCBR is 16-bit registers, used only in complementary PWM mode, that function as buffer registers for TCDR. The TCBR value is transferred to TCDR with the transfer timing set in TMDR. The TCBR value after reset is FFFFh.

22.2.26 Timer Interrupt Skipping Set Registers (TITCR)

Address(es): MTU.TITCR 0008 8630h

	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, see Table 22.40.	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, see Table 22.41.	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 TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the skipping counter (TITCNT).

T4VCOR[2:0] Bits (TCIV4 Interrupt Skipping Count Setting)
T3ACOR[2:0] Bits (TGIA3 Interrupt Skipping Count Setting)

These bits specify the TCIV3 and TGIA4 interrupt skipping count within the range from 0 to 7. For details, see Table 22.40 and Table 22.41.

Table 22.40 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

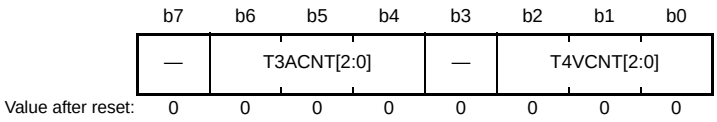
Bit 2	Bit 1	Bit 0	
T4VCOR2	T4VCOR1	T4VCOR0	Description
0	0	0	Does not perform TCIV4 interrupt skipping.
0	0	1	Sets the TCIV4 interrupt skipping count to 1.
0	1	0	Sets the TCIV4 interrupt skipping count to 2.
0	1	1	Sets the TCIV4 interrupt skipping count to 3.
1	0	0	Sets the TCIV4 interrupt skipping count 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 22.41 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T3ACOR2	T3ACOR1	T3ACOR0	Description
0	0	0	Does not perform TGIA3 interrupt skipping.
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.

22.2.27 Timer Interrupt Skipping Counters (TITCNT)

Address(es): MTU.TITCNT 0008 8631h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCNT[2:0]	TCIV4 Interrupt Counter	While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs.	R
b3	—	TCIV4 Interrupt Counter	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 TITCR 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: • To clear the TITCNT, clear the T3AEN and T4VEN bits in TITCR to 0.

TITCNT is 8-bit readable/writable counters. The MTU2 has two TITCNT counter. TITCNT retain their values even after stopping the count operation of MTU4.TCNT and MTU3.TCNT.

T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the T4VCNT[2:0] bits in TITCNT match the T4VCR[2:0] bits in TITCR
- When the T4VEN bit in TITCR is cleared to 0
- When the T4VCOR[2:0] bits in TITCR are cleared to 000b

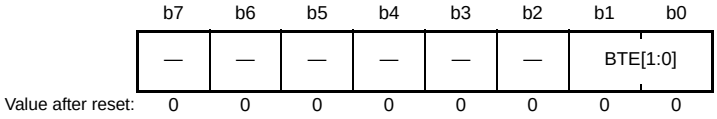
T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the T3ACNT[2:0] bits in TITCNT match the T3ACOR[2:0] bits in TITCR
- When the T3AEN bit in TITCR is cleared to 0
- When the TITCR.T3ACOR[2:0] bits are cleared to 000b

22.2.28 Timer Buffer Transfer Set Registers (TBTER)

Address(es): MTU.TBTER 0008 8632h



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 used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. See Table 22.42 for details.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TBTER is 8-bit readable/writable registers that 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 operation.

Table 22.42 Setting of TBTER.BTE[1:0] Bits

Bit 1	Bit 0	
BTE1	**BTE0**	**Description**
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping operation.
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 operation.*2
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 22.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the skipping count set bits (T3ACOR and T4VCOR) in TITCR 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 (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed

22.2.29 Timer Dead Time Enable Registers (TDER)

Address(es): MTU.TDER 0008 8634h

	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) *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDR must be set to 1 or a larger value.
Note 2. In complementary PWM mode, this bit must be read as 0 before writing 1 to it. No restrictions apply to writing 0 to the bit. Other than in complementary PWM mode, writing 1 to this bit is prohibited; when writing, write 0 to the bit.

TDERA is 8-bit readable/writable registers that control dead time generation in complementary PWM mode. The MTU3 has one TDER register. TDER 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 bit after reading TDER bit = 1

22.2.30 Timer Waveform Control Registers (TWCR)

Address(es): MTU.TWCR 0008 8660h

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

Note 1. Do not set to 1 unless complementary PWM mode is 1.

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Initial Output Inhibition Enable	0: Initial value specified in TOCR is output 1: Initial output is inhibited	R/(W) *1
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE	Compare Match Clear Enable	0: Counters are not cleared at MTU3.TGRA compare match 1: Counters are cleared at MTU3.TGRA compare match	R/(W)

Note 1. In complementary PWM mode, this bit must be read as 0 before writing 1 to it. No restrictions apply to writing 0 to the bit. Other than in complementary PWM mode, writing 1 to this bit is prohibited; when writing, write 0 to the bit.

TWCR is 8-bit readable/writable registers. TWCR controls the output waveform when synchronous counter clearing occurs in MTU3.TNCT and MTU4.TNCT in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA compare match.
The CCE bit and WRE bit in TWCR should be modified only while TCNT stops.

WRE Bit (Initial Output Inhibition Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.
The initial output is prohibited 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 value specified in TOCR is output regardless of the TWCR.WRE bit setting. The initial value specified in TOCR is also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT start operation.
For the Tb interval at the trough in complementary PWM mode, see Figure 22.40.
[Setting condition]

- When 1 is written to TWCR.WRE bit after reading TWCR.WRE bit = 0

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at TGRA3 compare match in complementary PWM mode.
[Setting condition]

- When 1 is written to CCE bit after reading CCE bit = 0

22.2.31 Noise Filter Control Registers (NFCR)

- NFCR (MTU0 to MTU4)

Address(es): MTU0.NFCR 0008 8690h, MTU1.NFCR 0008 8691h, MTU2.NFCR 0008 8692h, MTU3.NFCR 0008 8693h, MTU4.NFCR 0008 8694h

	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 Bit	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 Bit	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 Bit	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 Bit	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	00: PCLK/1 01: PCLK/8 10: PCLK/32 11: PCLK/Source that drives 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 NFCRs for MTU1 and MTU2. These bits are read as 0, and writing to them is not possible.

MTUn.NFCR is 8-bit readable and writable register (n = 0 to 4). These registers control enabling and disabling of the noise filters for the MTIOCnm (n = 0 to 4; m = A to D) pins and sets the sampling clocks for the noise filters.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. 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 and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. 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 and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. 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 and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

NFDEN Bit (Noise Filter D Enable)

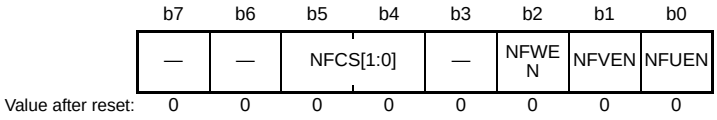
This bit disables or enables the noise filter for input from the MTIOCnD pin. Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

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, selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input-capture function.

- NFCR (MTU5)

Address(es): MTU5.NFCR 0008 8695h



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	00: PCLK/1 01: PCLK/8 10: PCLK/32 11: PCLK/Source that drives counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MTU5.NFCR is 8-bit readable and writable register. This register controls enabling and disabling of the noise filters for the MTIC5m (m = U, V, W) pins and sets the sampling clock for the noise filters.

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 NFUEN is changed, select the compare-match function in the timer I/O control register before changing the value.

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 NFVEN is changed, select the compare-match function in the timer I/O control register before changing the value.

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 NFWEN is changed, select the compare-match function in the timer I/O control register before changing the value.

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.

22.2.32 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA/B), and timer A/D converter start request cycle set buffer registers (TADCOBRA/B) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Always access the registers in 16-bit units.
All registers other than the above registers are 8-bit registers, so read/write access should be performed in 8-bit units.

22.3 Operation

22.3.1 Basic Functions

Each channel has TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR can be used as an input capture register or an output compare register.

(1) Counter Operation

When one of bits CST0 to CST4 in TSTR or bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR 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 22.4 shows an example of the count operation setting procedure.

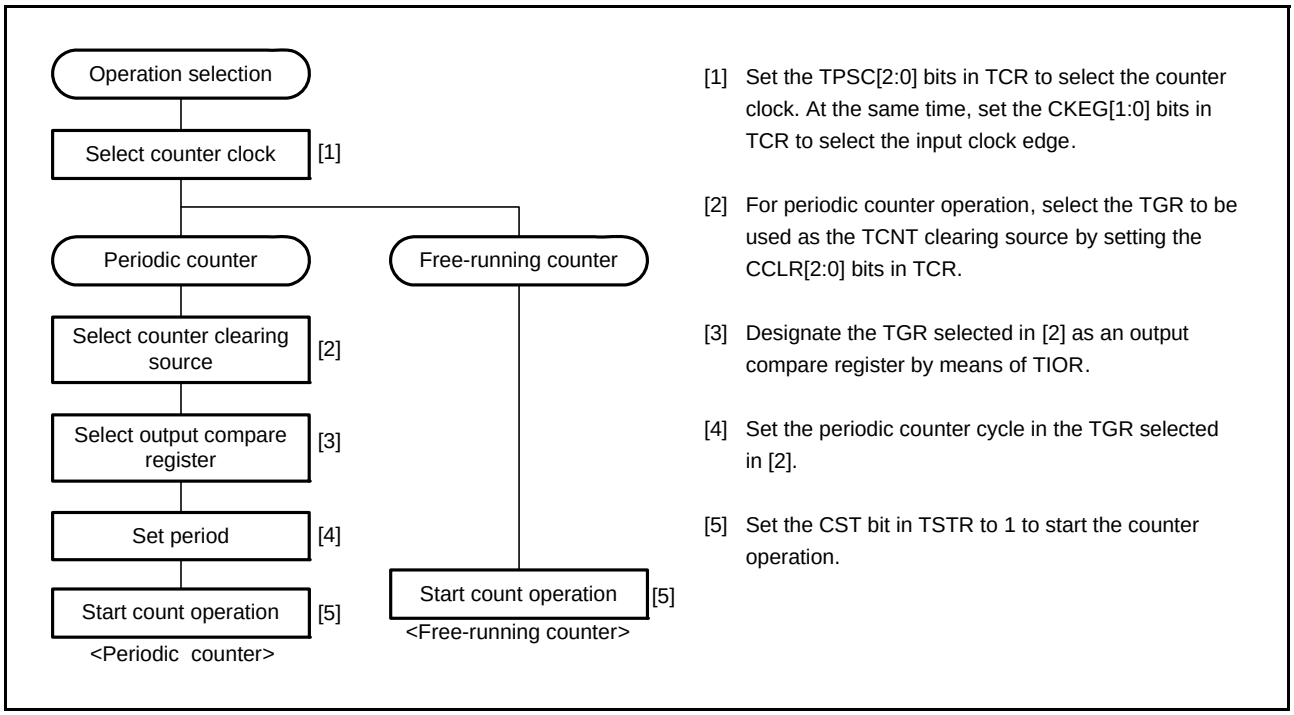


Figure 22.4 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant bit in 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), the MTU requests an interrupt if the corresponding TCIEV bit in TIER is 1. After an overflow, TCNT starts counting up again from 0000h.
Figure 22.5 illustrates free-running counter operation.

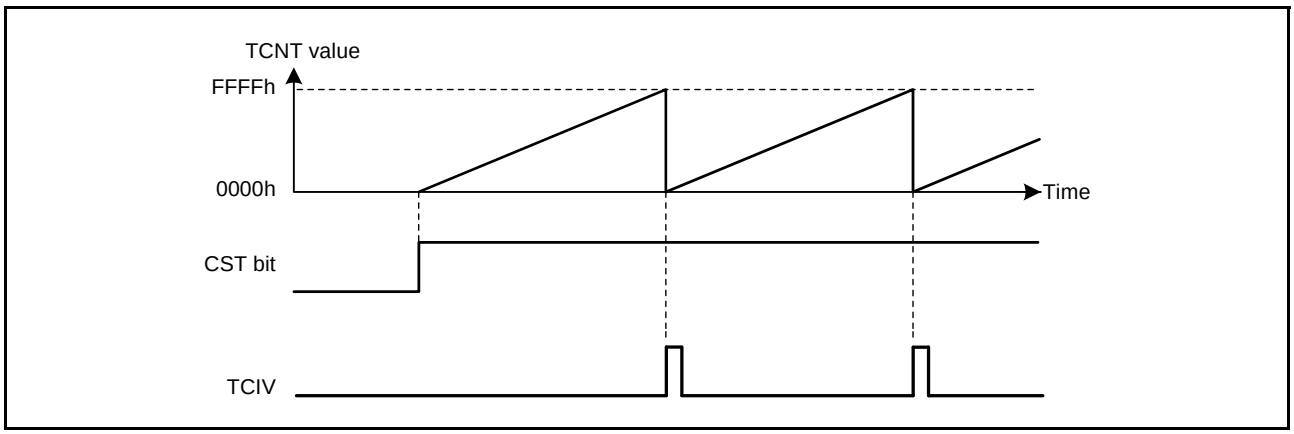


Figure 22.5 Free-Running Counter Operation

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 TSTR is set to 1. When the count matches the value in TGR, TCNT is cleared to 0000h.
If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0000h.
Figure 22.6 illustrates periodic counter operation.

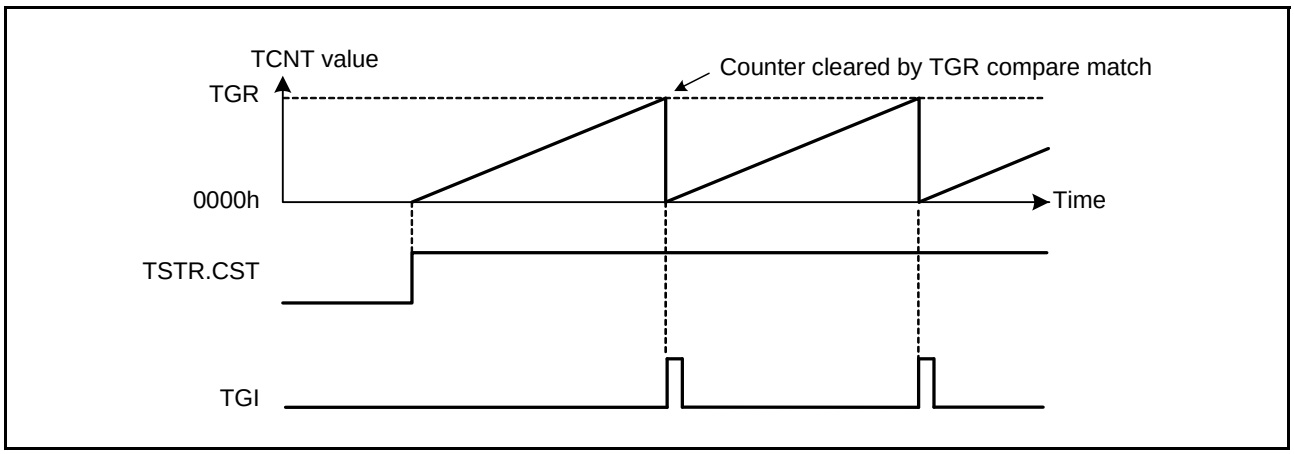


Figure 22.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU can output low or high or toggle output from the corresponding output pin using compare match.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 22.7 shows an example of the procedure for setting waveform output by compare match

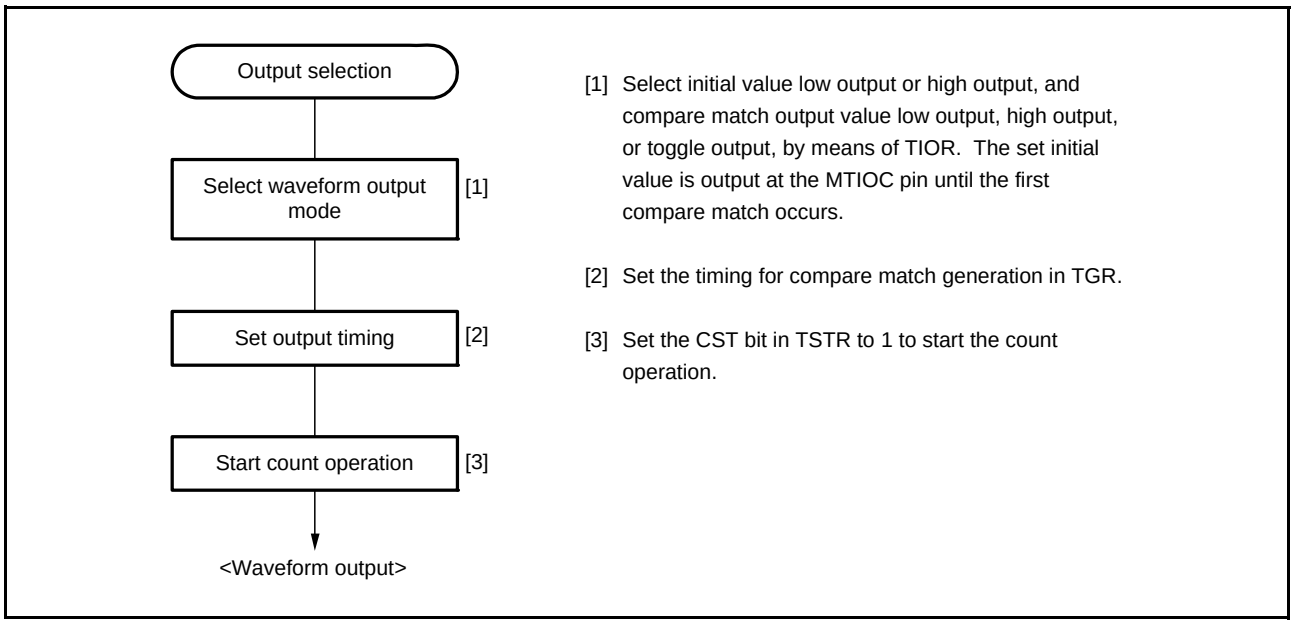


Figure 22.7 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 22.8 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.

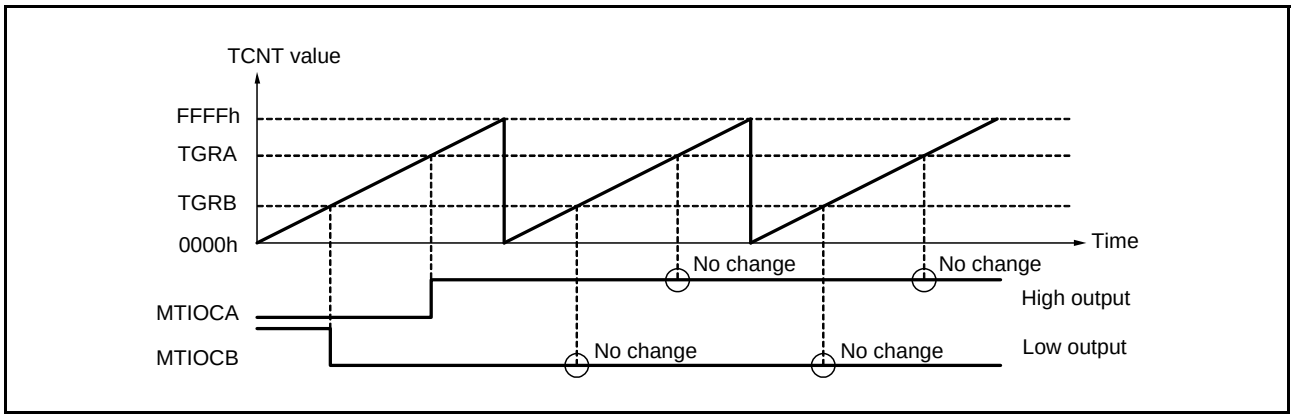


Figure 22.8 Example of Low Output and High Output Operation

Figure 22.9 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.

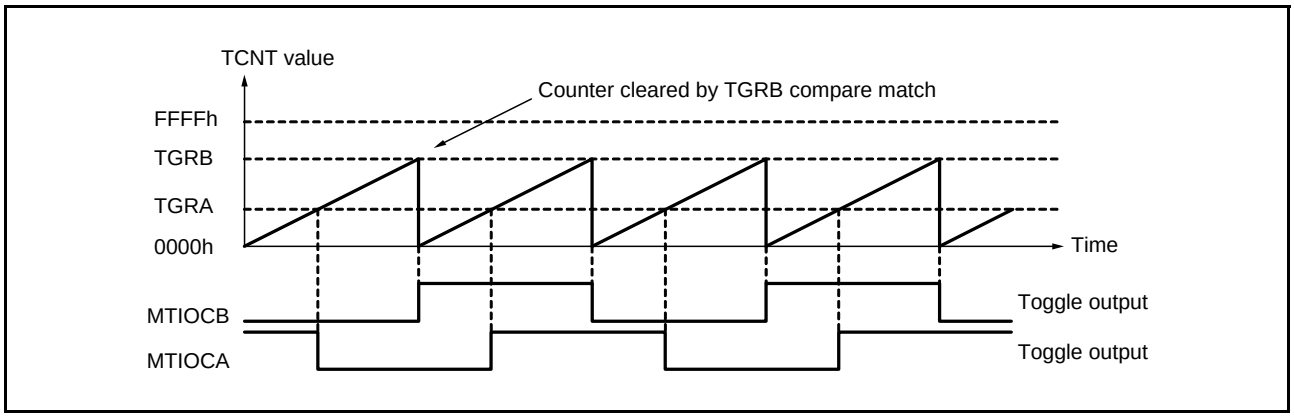


Figure 22.9 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOC pin input edge.
The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's counter input clock or compare match signal can also be specified as the input capture source.

Note: • When another channel's counter input clock is used as the input capture input for MTU0 and MTU1, PCLK/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 22.10 shows an example of the input capture operation setting procedure.

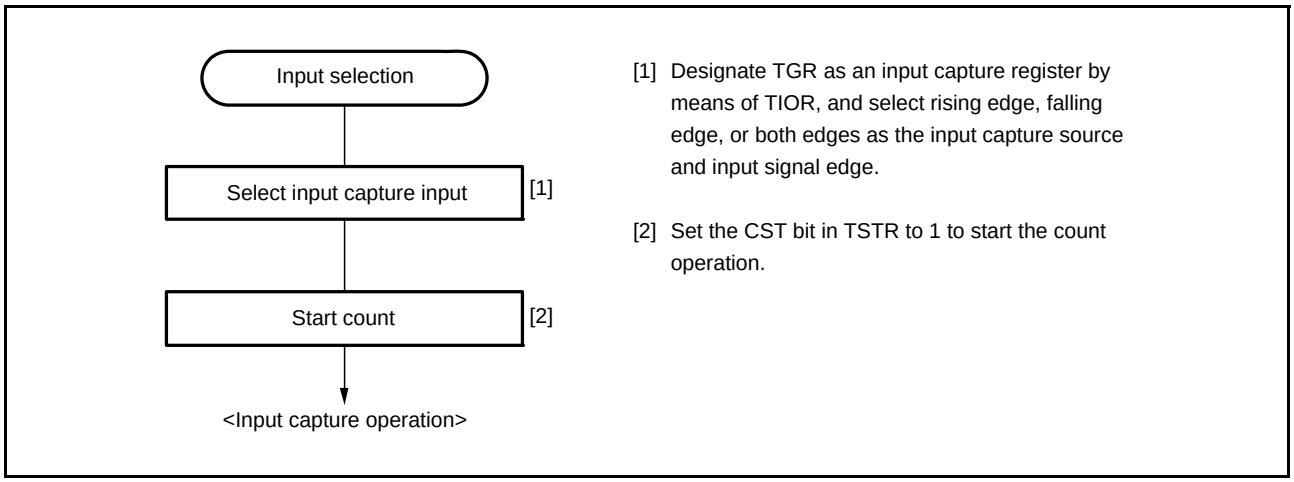


Figure 22.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 22.11 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.

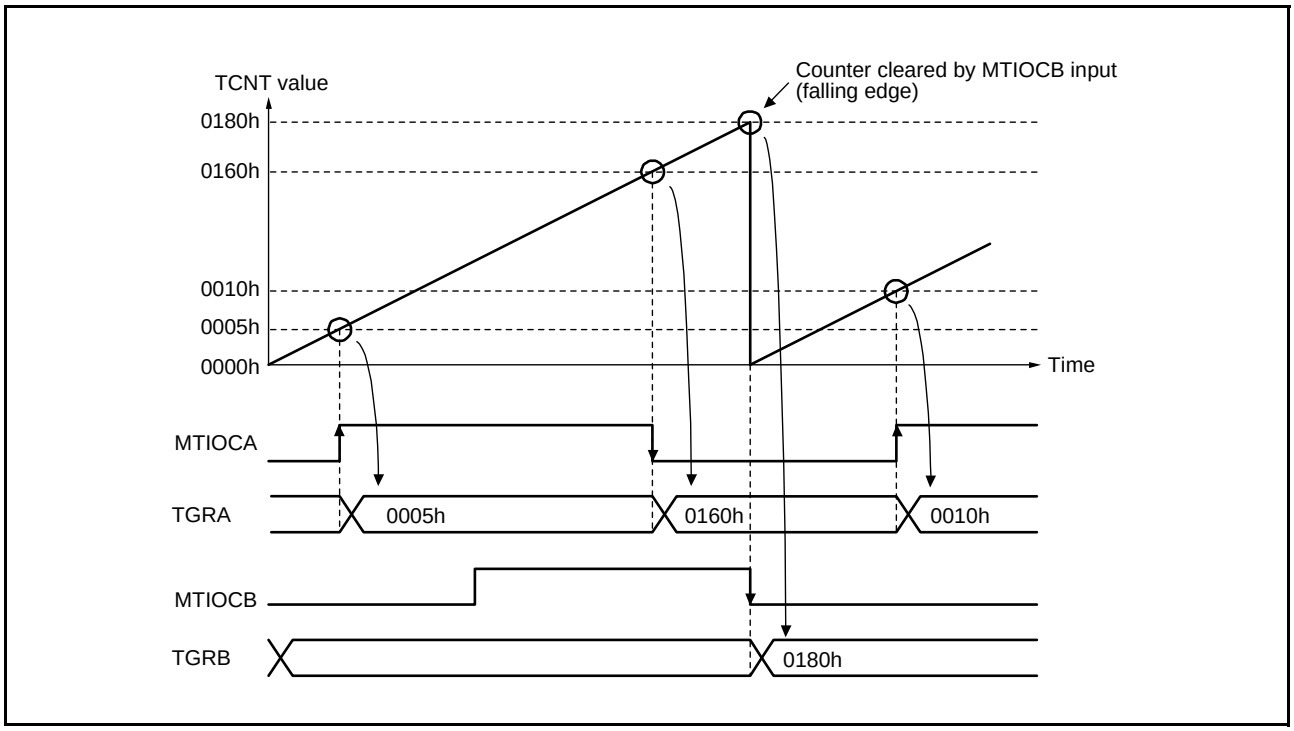


Figure 22.11 Example of Input Capture Operation

22.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous presetting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.
Synchronous operation increases the number of TGR registers assigned to a single time base.
MTU0 to MTU4 can all be designated for synchronous operation. MTU5 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 22.12 shows an example of the synchronous operation setting procedure.

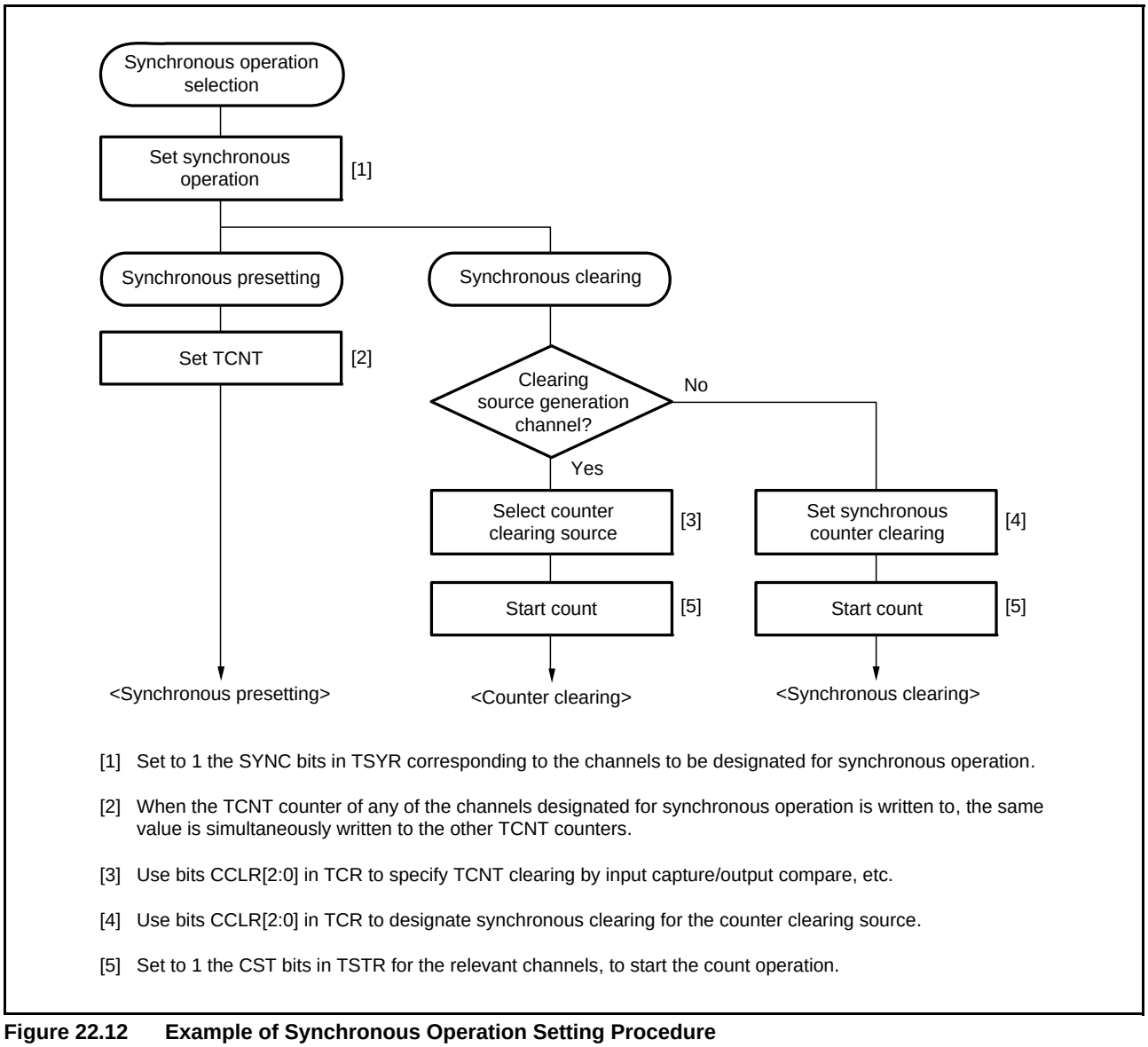


Figure 22.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 22.13 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU2, 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 presetting 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, see section 22.3.5, PWM Modes.

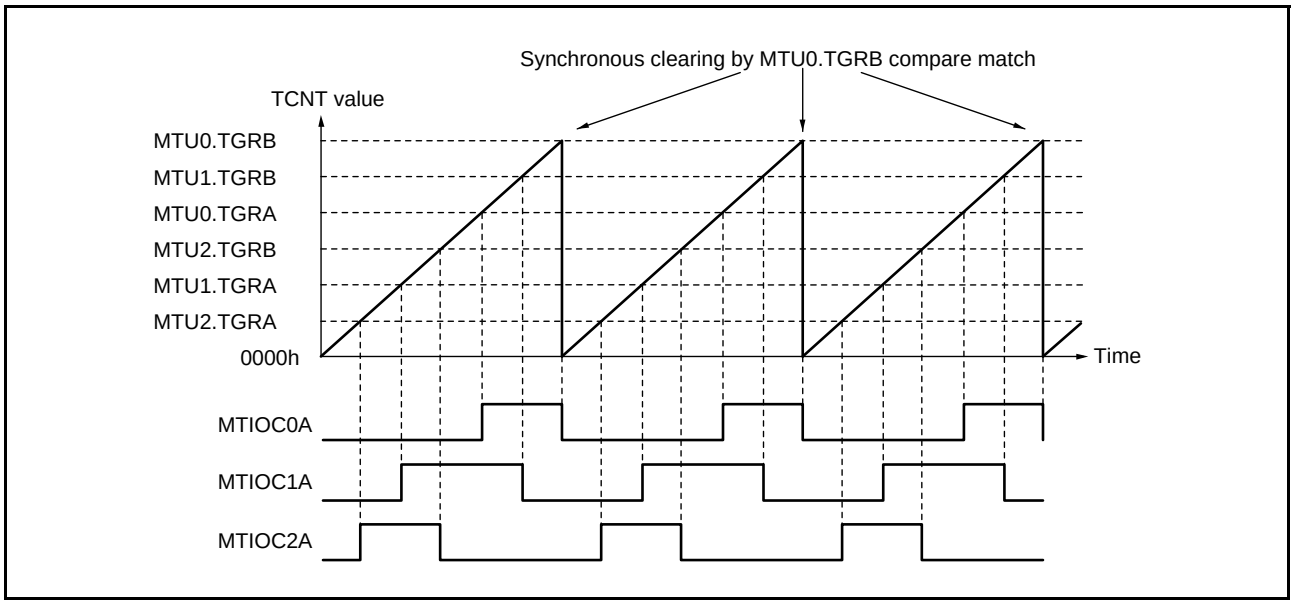


Figure 22.13 Example of Synchronous Operation

22.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, and MTU4, 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 22.43 lists the register combinations used in buffer operation.

Table 22.43 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

- 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 22.14.

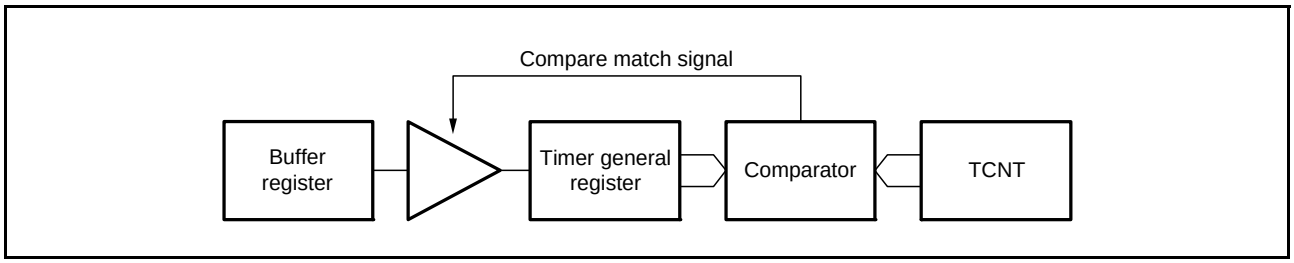


Figure 22.14 Compare Match Buffer Operation

- 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 22.15.

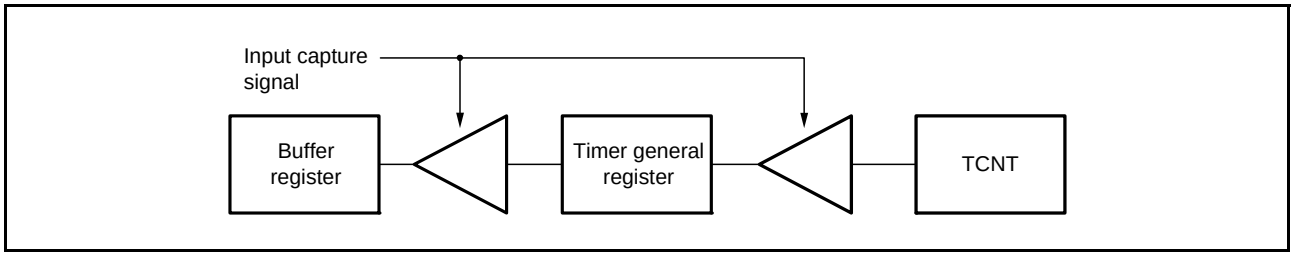


Figure 22.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 22.16 shows an example of the buffer operation setting procedure.

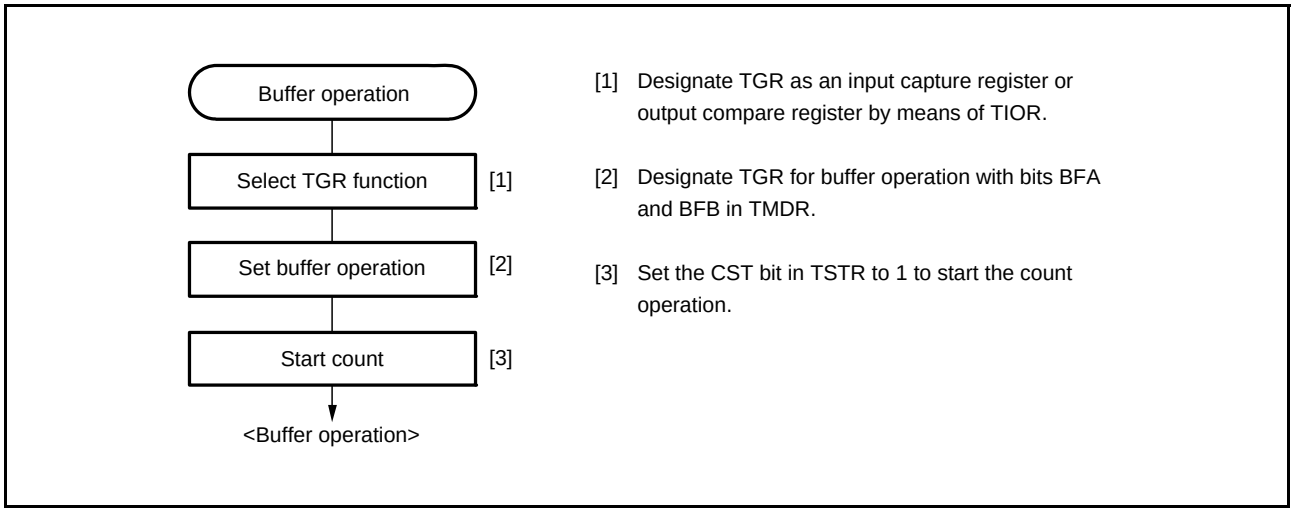


Figure 22.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 22.17 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, see section 22.3.5, PWM Modes.

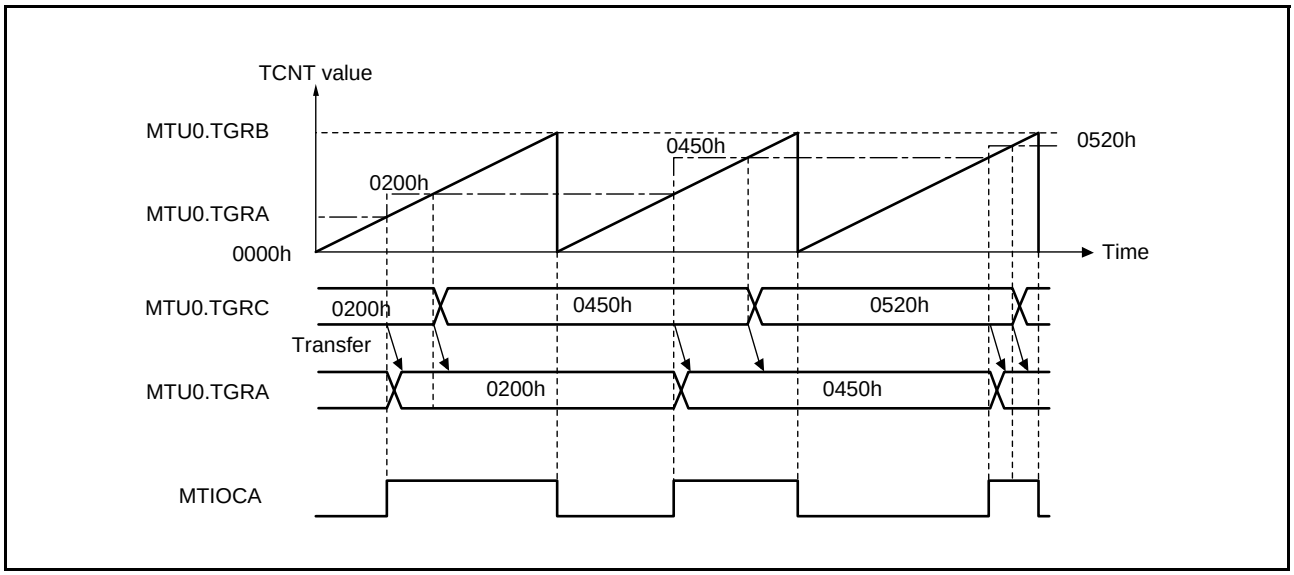


Figure 22.17 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 22.18 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.

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.

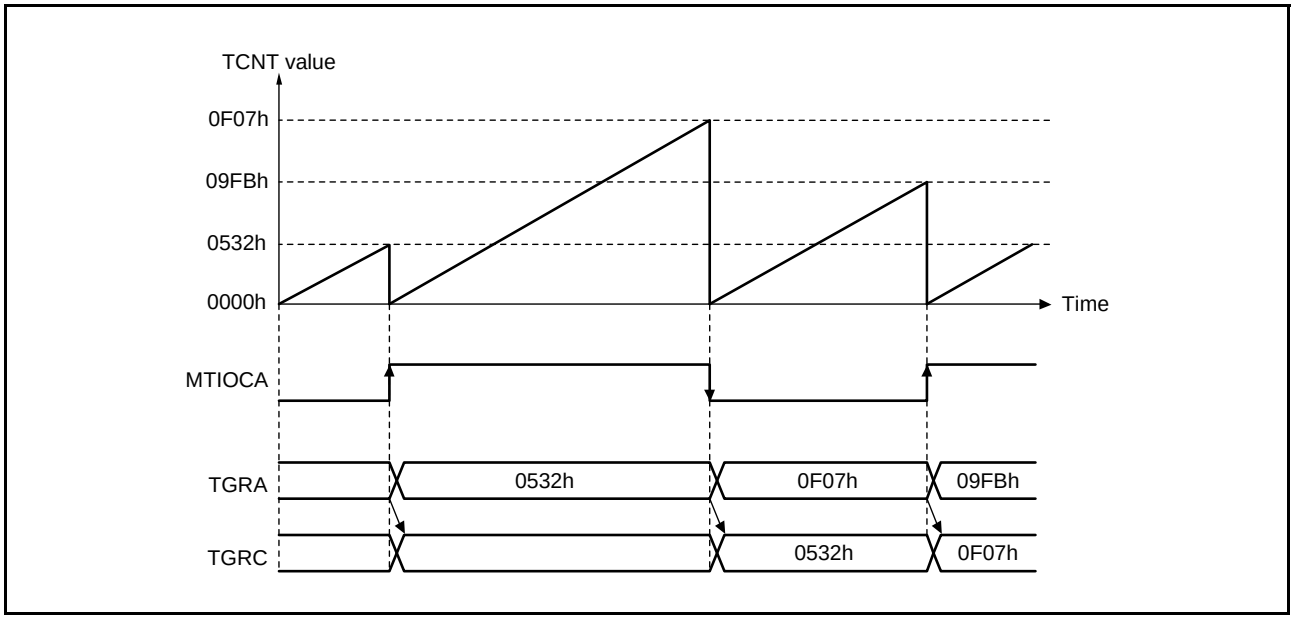


Figure 22.18 Example of Buffer Operation (2)

(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 and MTU4 by setting the buffer operation transfer mode registers (MTU0.TBTM, MTU3.TBTM, and MTU4.TBTM). Either compare match (initial setting) 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 → 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 22.19 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.

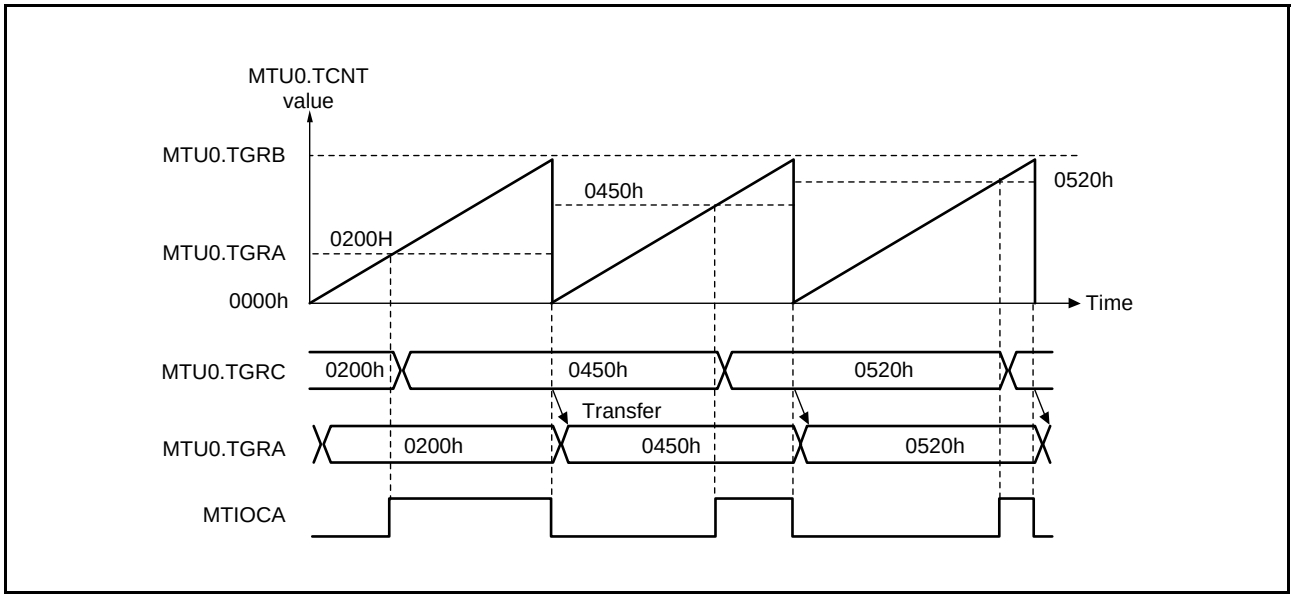


Figure 22.19 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC-to-MTU0.TGRA Transfer Timing

22.3.4 Cascaded Operation

In cascaded operation, 16-bit counters in different two channels are used together as a 32-bit counter.
This function works when overflow/underflow of MTU2.TCNT is selected as the counter clock for MTU1 through the TPSC[2:0] bits in TCR.
Underflow occurs only when the lower 16 bits of TCNT is in phase counting mode.
Table 22.44 lists the register combinations used in cascaded operation.

Note: • When phase counting mode is set for MTU1 or MTU2, the counter clock setting is invalid and the counters operate independently in phase counting mode.

Table 22.44 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). For input capture in cascade connection, refer to section 22.6.22, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection. Table 22.45 lists the TICCR setting and input capture input pins.

Table 22.45 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 22.20 shows an example of the cascaded operation setting procedure.

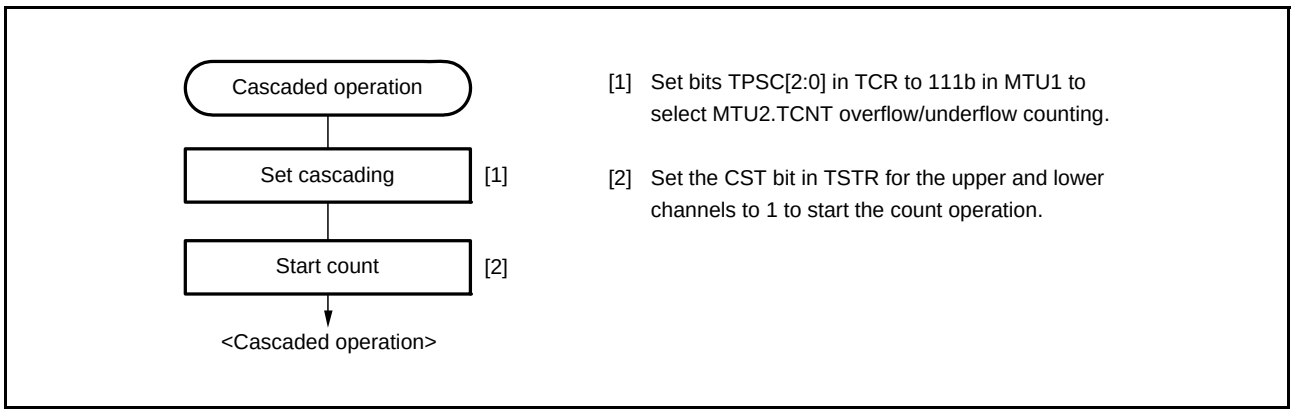


Figure 22.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 22.21 illustrates the operation when MTU2.TCNT overflow/underflow counting has been set for MTU1.TCNT and phase counting mode has been designated for MTU2.
MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

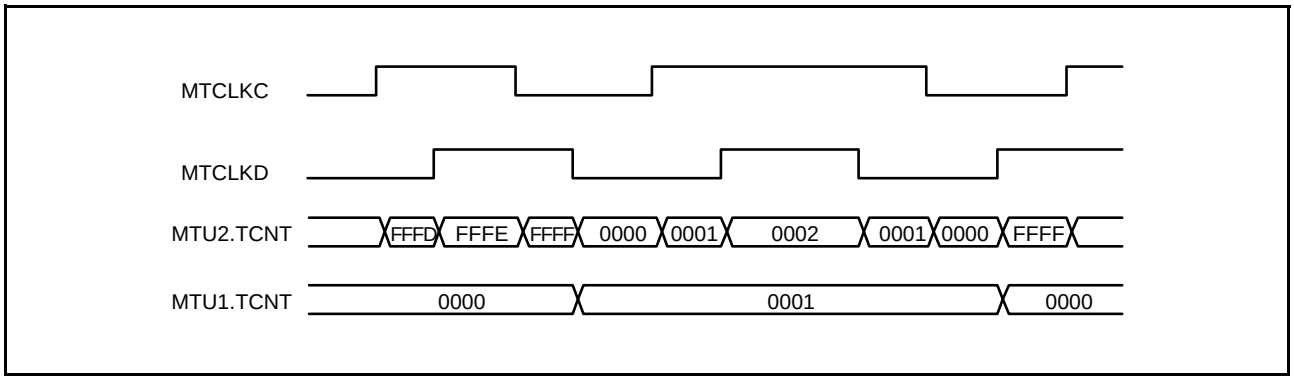


Figure 22.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 22.22 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 IOA3 to IOA0 bits in MTU1.TIOR have selected the MTIOC1A rising edge for the input capture timing while the IOA3 to IOA0 bits in MTU2.TIOR 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.

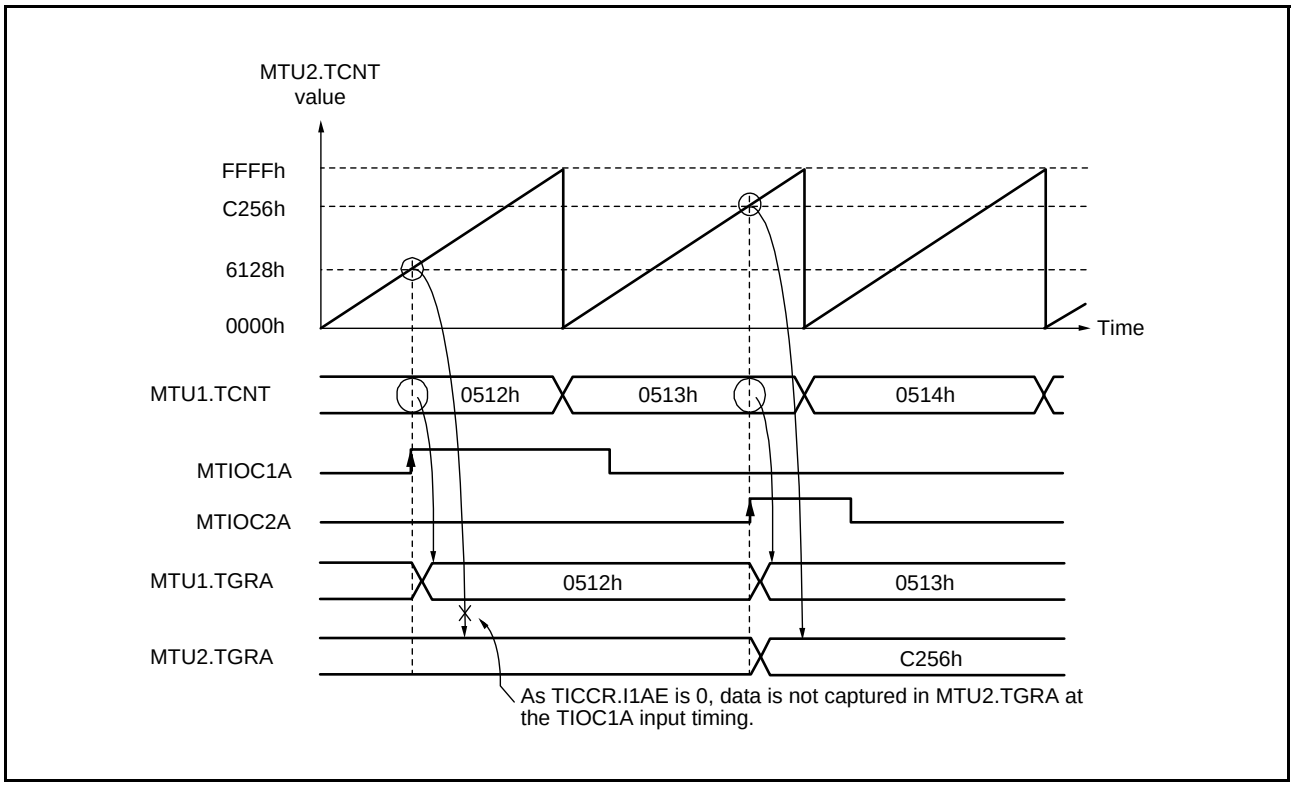


Figure 22.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 22.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE and I1AE bits in TICCR 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 IOA3 to IOA0 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 OR result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.

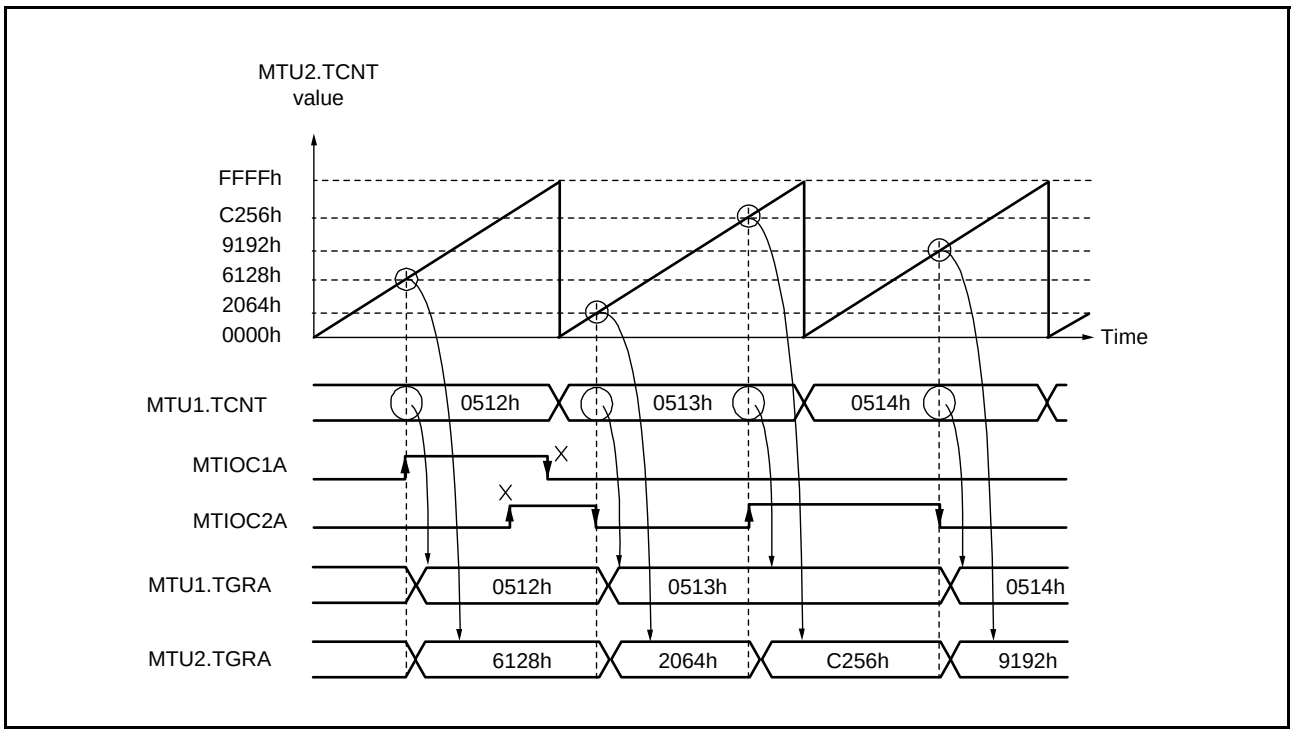


Figure 22.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 22.24 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 IOA3 to IOA0 bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA3 to IOA0 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.

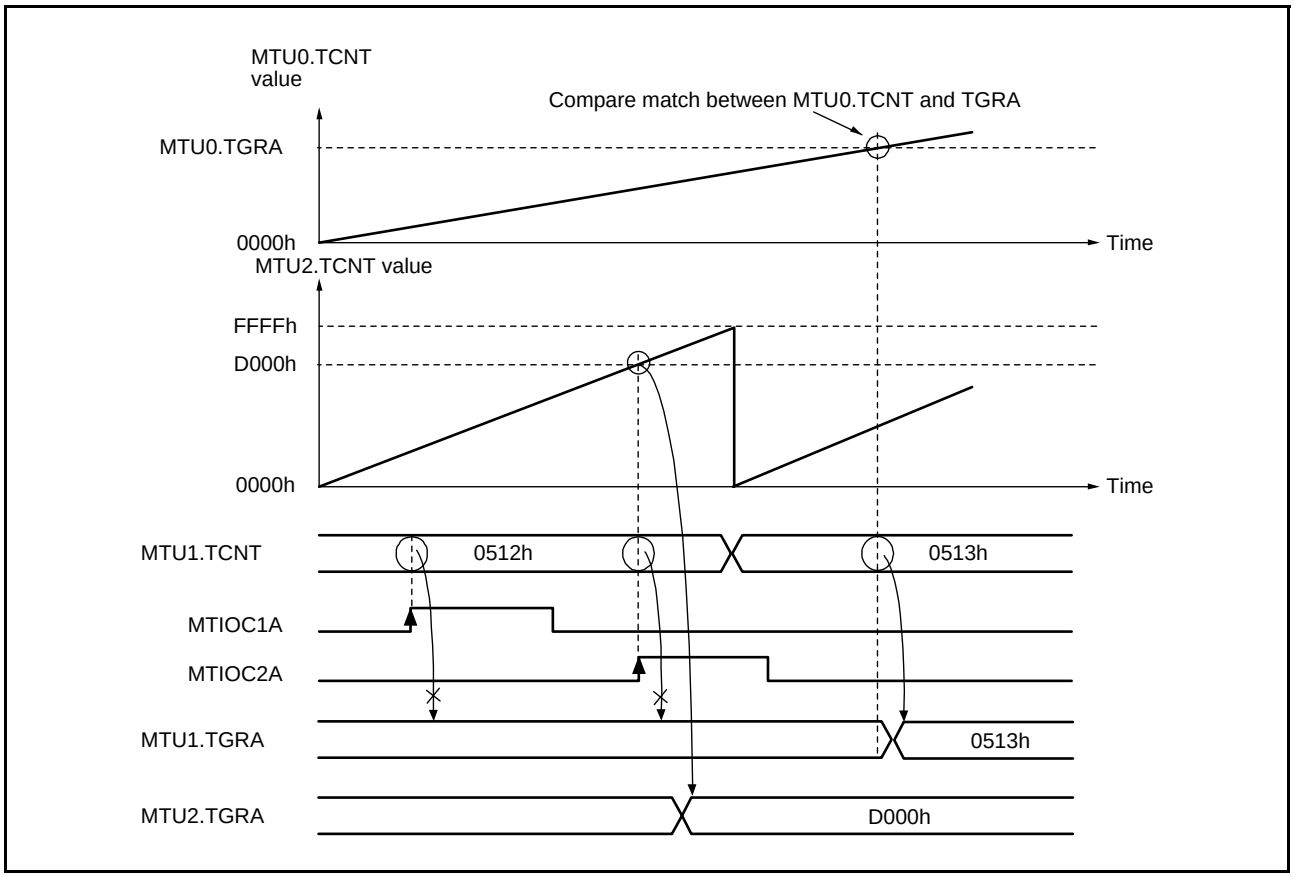


Figure 22.24 Cascaded Operation Example (d)

22.3.5 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 can be set to PWM mode independently. Synchronous operation is also possible.
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 bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare matches B and D. 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, up to eight phases of PWM waveforms can be output.

(b) PWM Mode 2

PWM 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 synchronized 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.

In PWM mode 2, up to eight phases of PWM waveforms can be output when using synchronous operation in combination.
The correspondence between PWM output pins and registers is listed in Table 22.46.

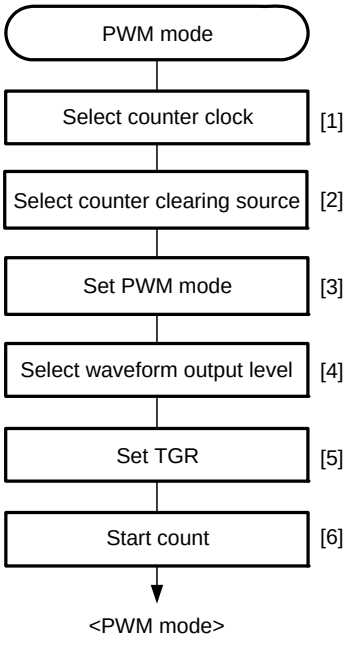
Table 22.46 PWM Output Registers and Output Pins

Channel	Register	Output Pins	
		PWM Mode 1	PWM Mode 2
MTU0	MTU0.TGRA	MTIOC0A	MTIOC0A
	MTU0.TGRB		MTIOC0B
	MTU0.TGRC	MTIOC0C	MTIOC0C
	MTU0.TGRD		MTIOC0D
MTU1	MTU1.TGRA	MTIOC1A	MTIOC1A
	MTU1.TGRB		MTIOC1B
MTU2	MTU2.TGRA	MTIOC2A	MTIOC2A
	MTU2.TGRB		MTIOC2B
MTU3	MTU3.TGRA	MTIOC3A	Setting prohibited
	MTU3.TGRB		
	MTU3.TGRC	MTIOC3C	
	MTU3.TGRD		
MTU4	MTU4.TGRA	MTIOC4A	
	MTU4.TGRB		
	MTU4.TGRC	MTIOC4C	
	MTU4.TGRD		

Note: • In PWM mode 2, PWM output is not possible for the TGR register in which the PWM cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 22.25 shows an example of the PWM mode setting procedure.



[1] Set the TPSC[2:0] bits in TCR to select the counter clock. At the same time, set the CKEG[1:0] bits in TCR to select the input clock edge.

[2] Set the CCLR[2:0] bits in TCR to select the TGR to be used as the TCNT clearing source.

[3] Select the PWM mode with bits MD[3:0] in TMDR.

[4] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.

[5] Set the cycle in the TGR selected in [2], and set the duty in the other TGR.

[6] Set the CST bit in TSTR to 1 to start the count operation.

Figure 22.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 22.26 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.

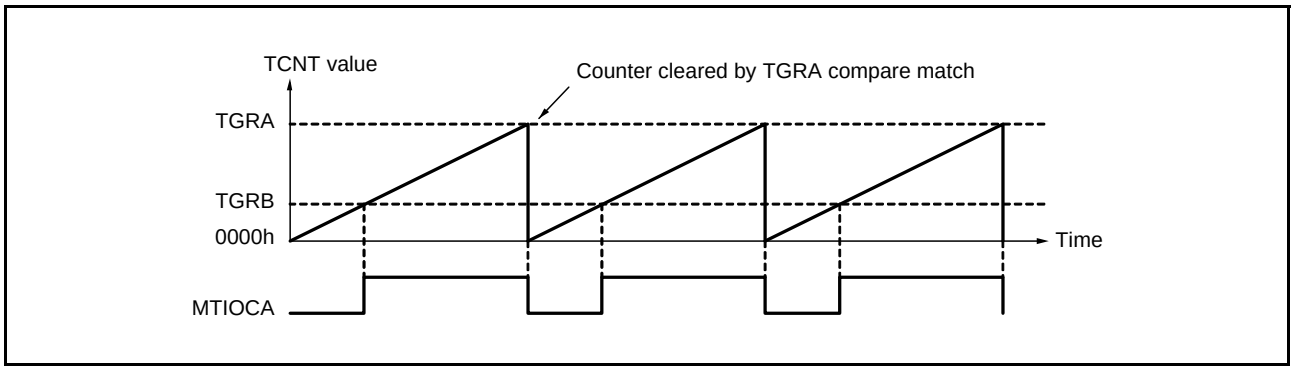


Figure 22.26 Example of PWM Mode Operation

Figure 22.27 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 0 is set as the initial output value and 1 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 TGR1B is used as the cycle, and the values set in the other TGRs are used as the duty.

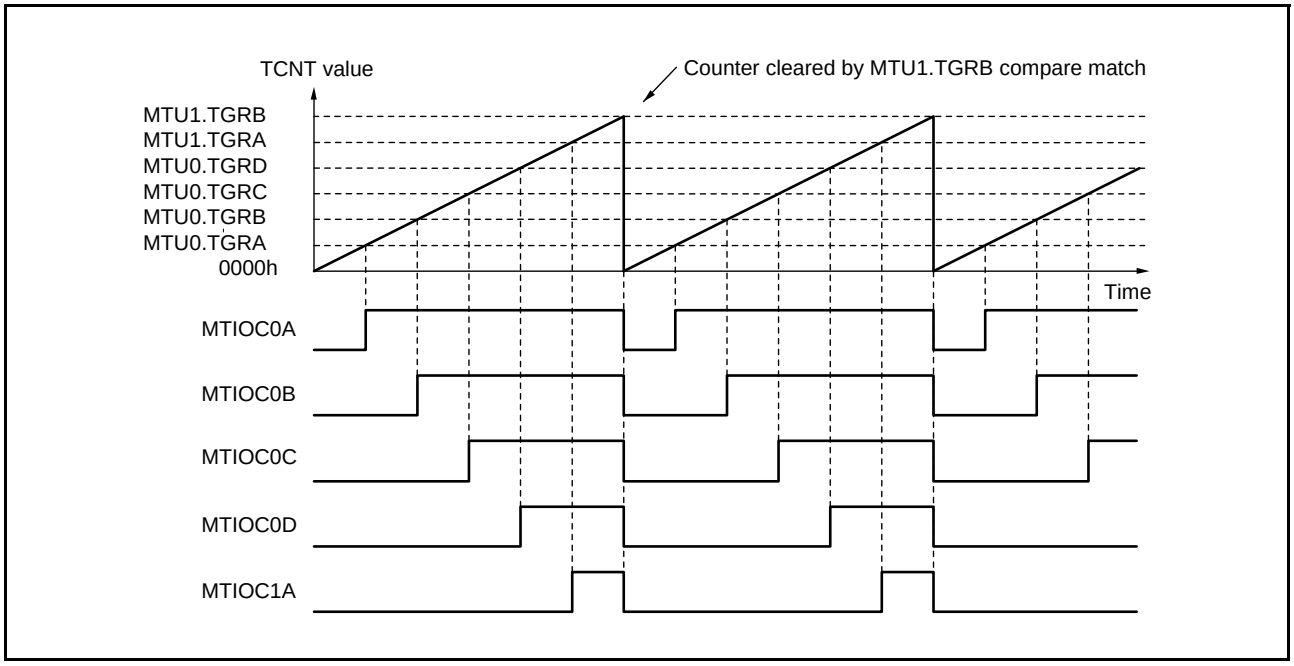


Figure 22.27 Example of PWM Mode Operation

Figure 22.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

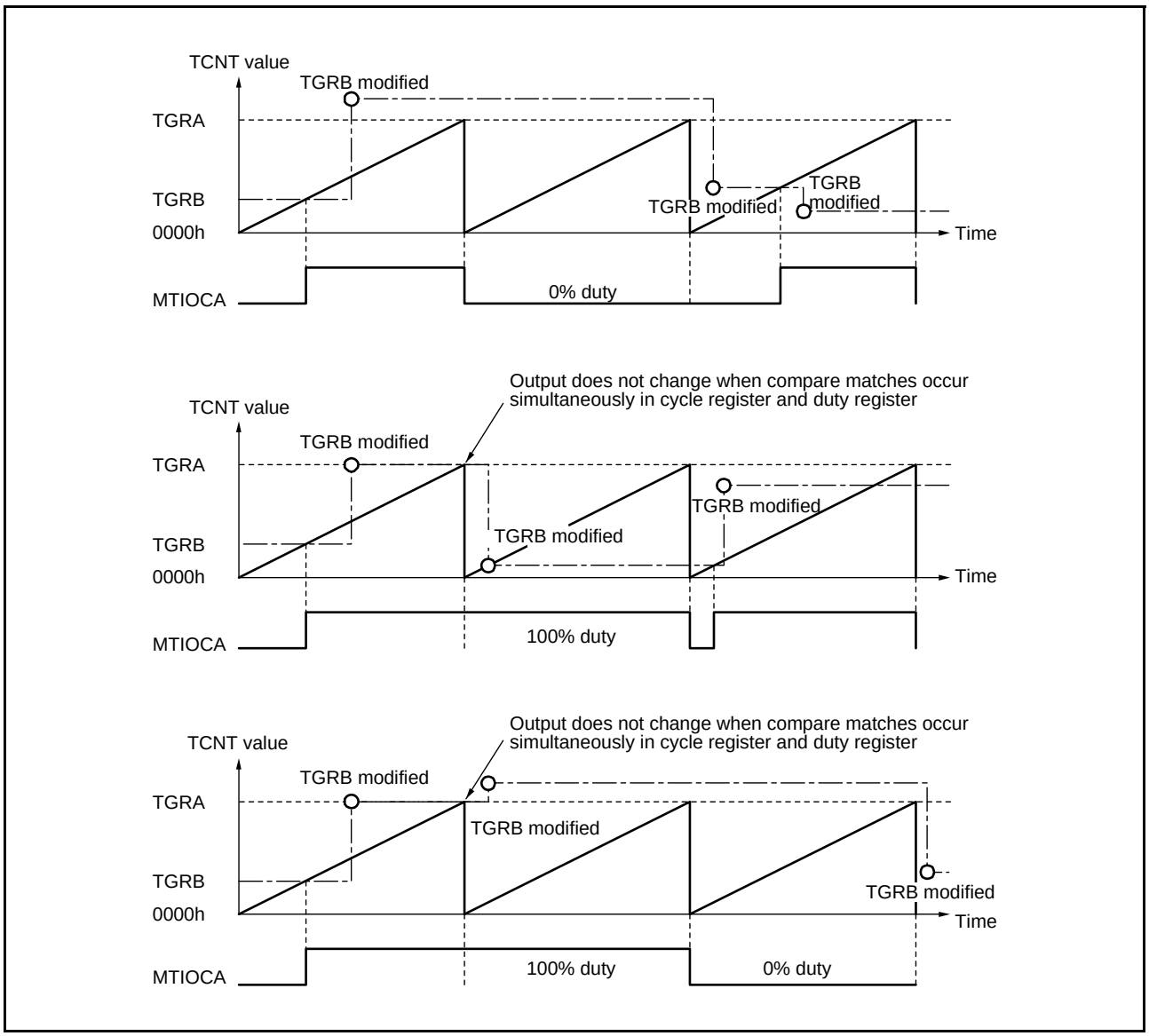


Figure 22.28 Examples of PWM Mode Operation

22.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external input clocks is detected and TCNT is incremented or decremented accordingly. This mode can be set for MTU1 and MTU2.

When phase counting mode is specified, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC[2:0] and bits CKEG[1:0] in TCR. However, the functions of bits CCLR[1:0] in TCR and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If an overflow occurs while TCNT is counting up, a TCIV interrupt is generated while the TCIEV bit in the corresponding TIER is 1. If an underflow occurs while TCNT is counting down, a TCIU interrupt is generated while the TCIEU bit in the corresponding TIER is 1.

The TCFD bit in TSR is the count direction flag. Read the TCFD flag to check whether TCNT is counting up or down.

Table 22.47 lists the correspondence between external clock pins and channels.

Table 22.47 Clock Input Pins in Phase Counting Mode

	External Clock Input Pins	
Channel	**A-Phase**	**B-Phase**
MTU1	MTCLKA	MTCLKB
MTU2	MTCLKC	MTCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 22.29 shows an example of the phase counting mode setting procedure.

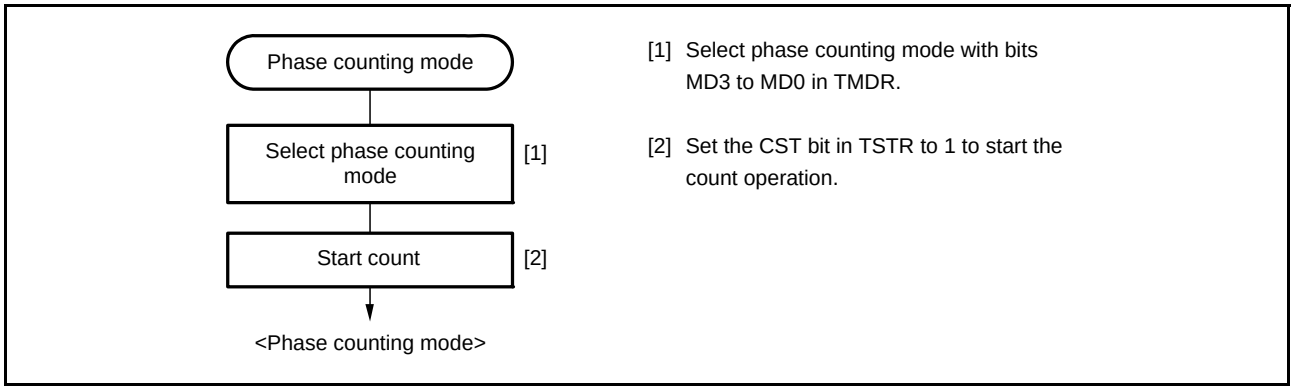


Figure 22.29 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are four modes according to the count conditions.

(a) Phase Counting Mode 1

Figure 22.30 shows an example of operation in phase counting mode 1, and Table 22.48 summarizes the TCNT up/down-count conditions.

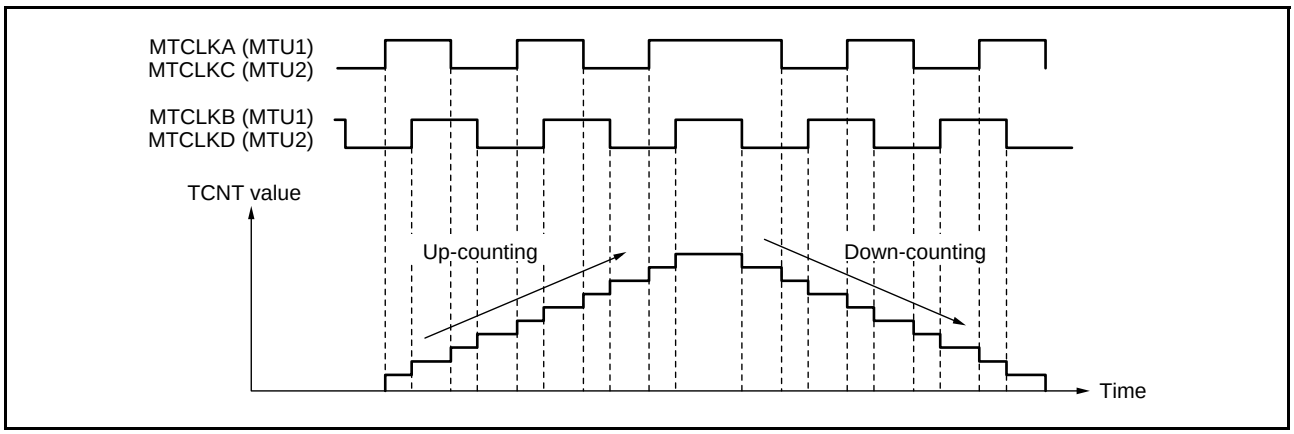


Figure 22.30 Example of Operation in Phase Counting Mode

Table 22.48 Up/Down-Count Conditions in Phase Counting Mode 1

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High level	↑	Up-counting
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-counting
Low level	↑	
↑	High level	
↓	Low level	

↑ : Rising edge

↓ : Falling edge

(b) Phase Counting Mode 2

Figure 22.31 shows an example of operation in phase counting mode 2, and Table 22.49 summarizes the TCNT up/down-count conditions.

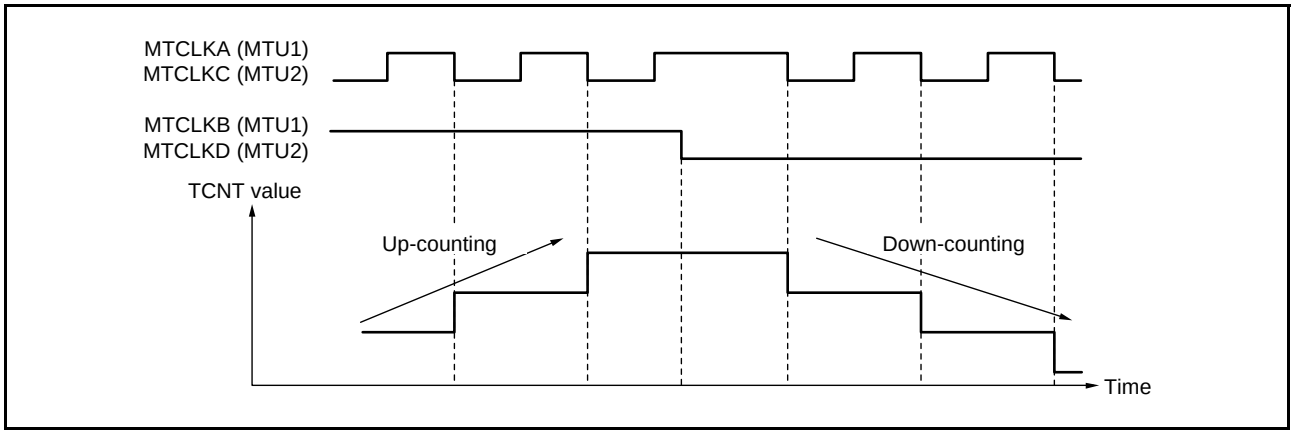


Figure 22.31 Example of Operation in Phase Counting Mode 2

Table 22.49 Up/Down-Count Conditions in Phase Counting Mode 2

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High level	↑	None (Don't care)
Low level	↓	None (Don't care)
↑	Low level	None (Don't care)
↓	High level	Up-counting
High level	↓	None (Don't care)
Low level	↑	None (Don't care)
↑	High level	None (Don't care)
↓	Low level	Down-counting

↑ : Rising edge

↓ : Falling edge

(c) Phase Counting Mode 3

Figure 22.32 shows an example of operation in phase counting mode 3, and Table 22.50 summarizes the TCNT up/down-count conditions.

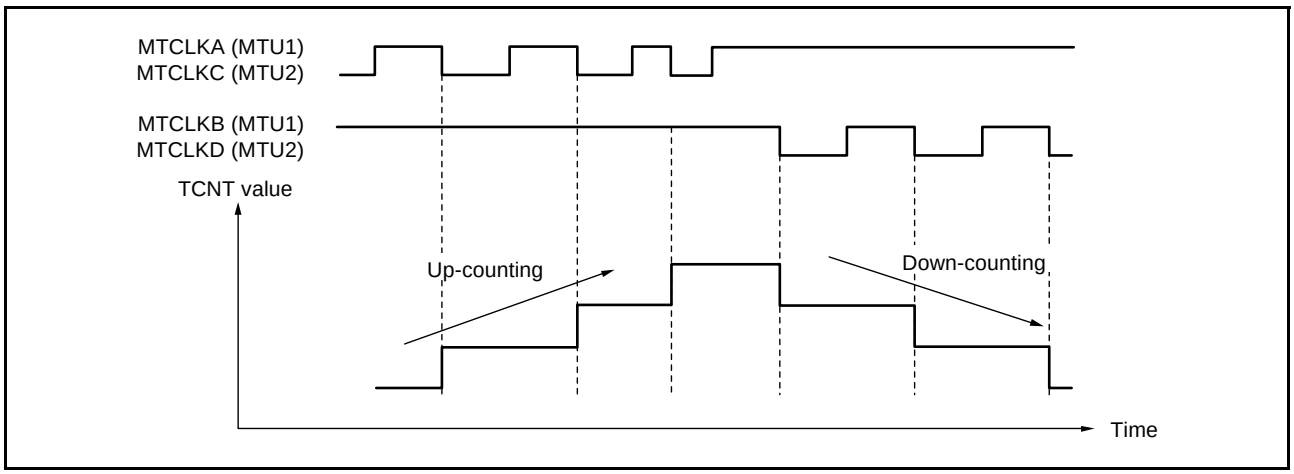


Figure 22.32 Example of Operation in Phase Counting Mode 3

Table 22.50 Up/Down-Count Conditions in Phase Counting Mode 3

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High level	↑	None (Don't care)
Low level	↓	None (Don't care)
↑	Low level	None (Don't care)
↓	High level	Up-counting
High level	↓	Down-counting
Low level	↑	None (Don't care)
↑	High level	None (Don't care)
↓	Low level	None (Don't care)

↑ : Rising edge

↓ : Falling edge

(d) Phase Counting Mode 4

Figure 22.33 shows an example of operation in phase counting mode 4, and Table 22.51 summarizes the TCNT up/down-count conditions.

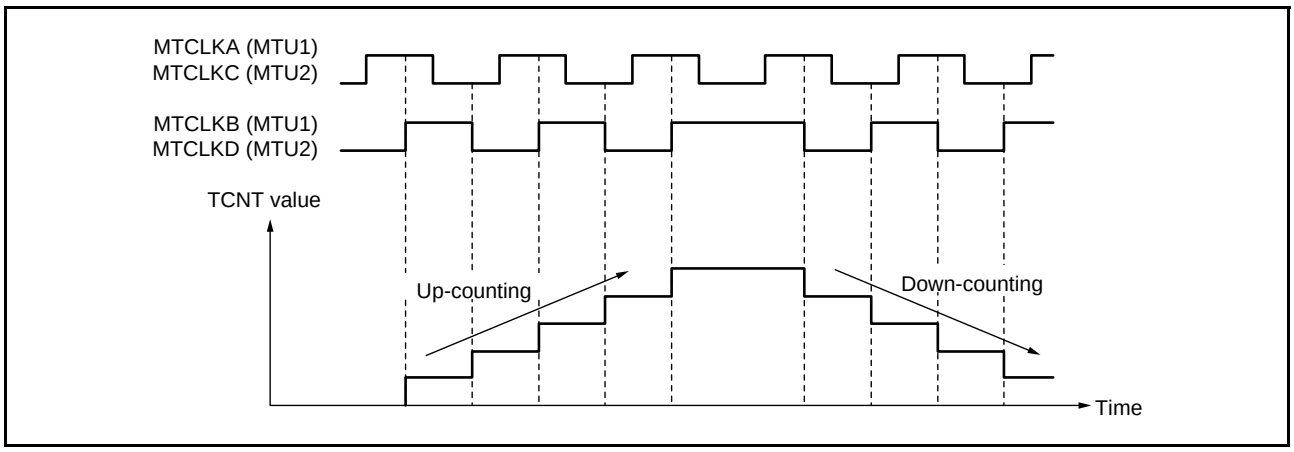


Figure 22.33 Example of Operation in Phase Counting Mode

Table 22.51 Up/Down-Count Conditions in Phase Counting Mode 4

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High level	↑	Up-counting
Low level	↓	
↑	Low level	None (Don't care)
↓	High level	
High level	↓	Down-counting
Low level	↑	
↑	High level	None (Don't care)
↓	Low level	

↑ : Rising edge

↓ : Falling edge

(3) Phase Counting Mode Application Example

Figure 22.34 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 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 TGRD operating in buffer mode. The MTU1 counter input 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 TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and 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.

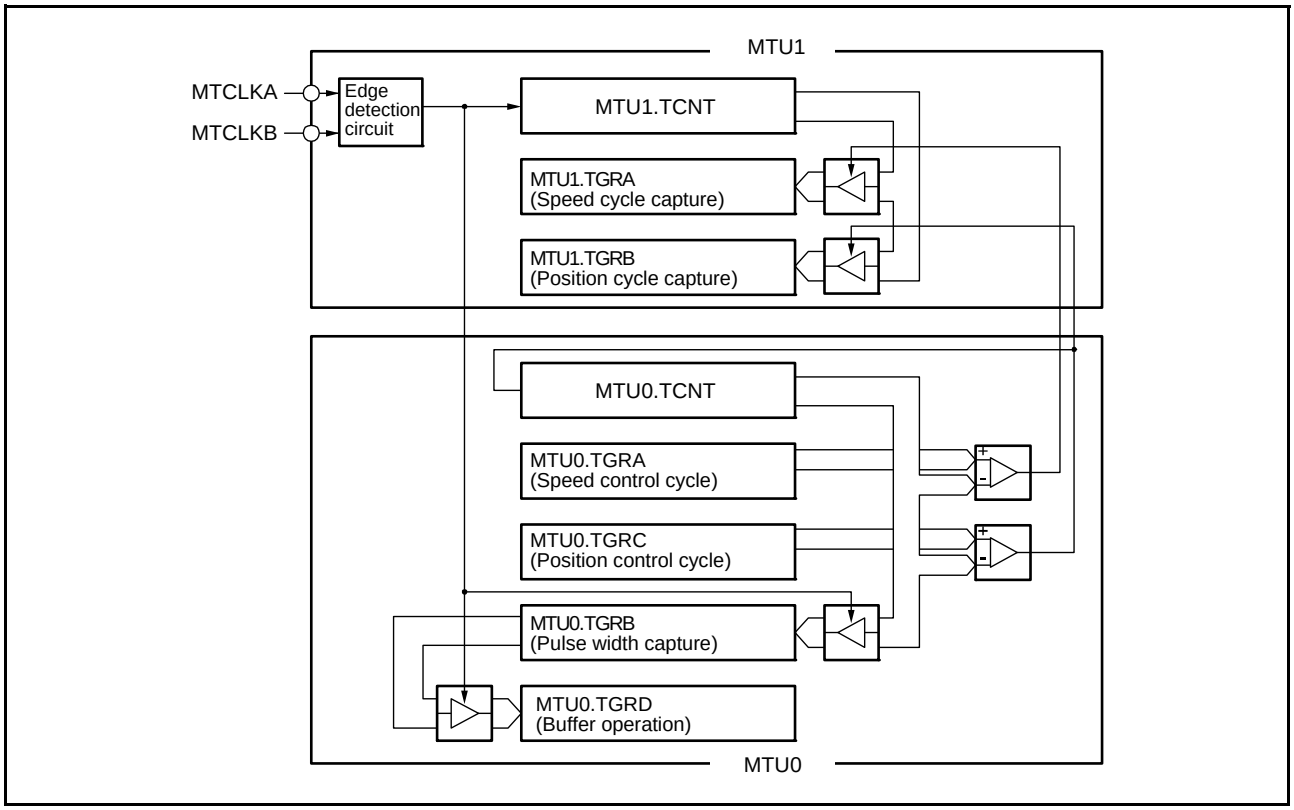


Figure 22.34 Phase Counting Mode Application Example

22.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three phases of positive and negative PWM waveforms that share a common wave transition point can be output by combining MTU3 and MTU4.
When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D pins function as PWM output pins and timer counter 3 (MTU3.TCNT) functions as an up-counter.
Table 22.52 lists the PWM output pins. Table 22.53 lists the settings of the registers.

Table 22.52 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)

Table 22.53 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

(1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 22.35 shows an example of procedure for setting the reset-synchronized PWM mode.

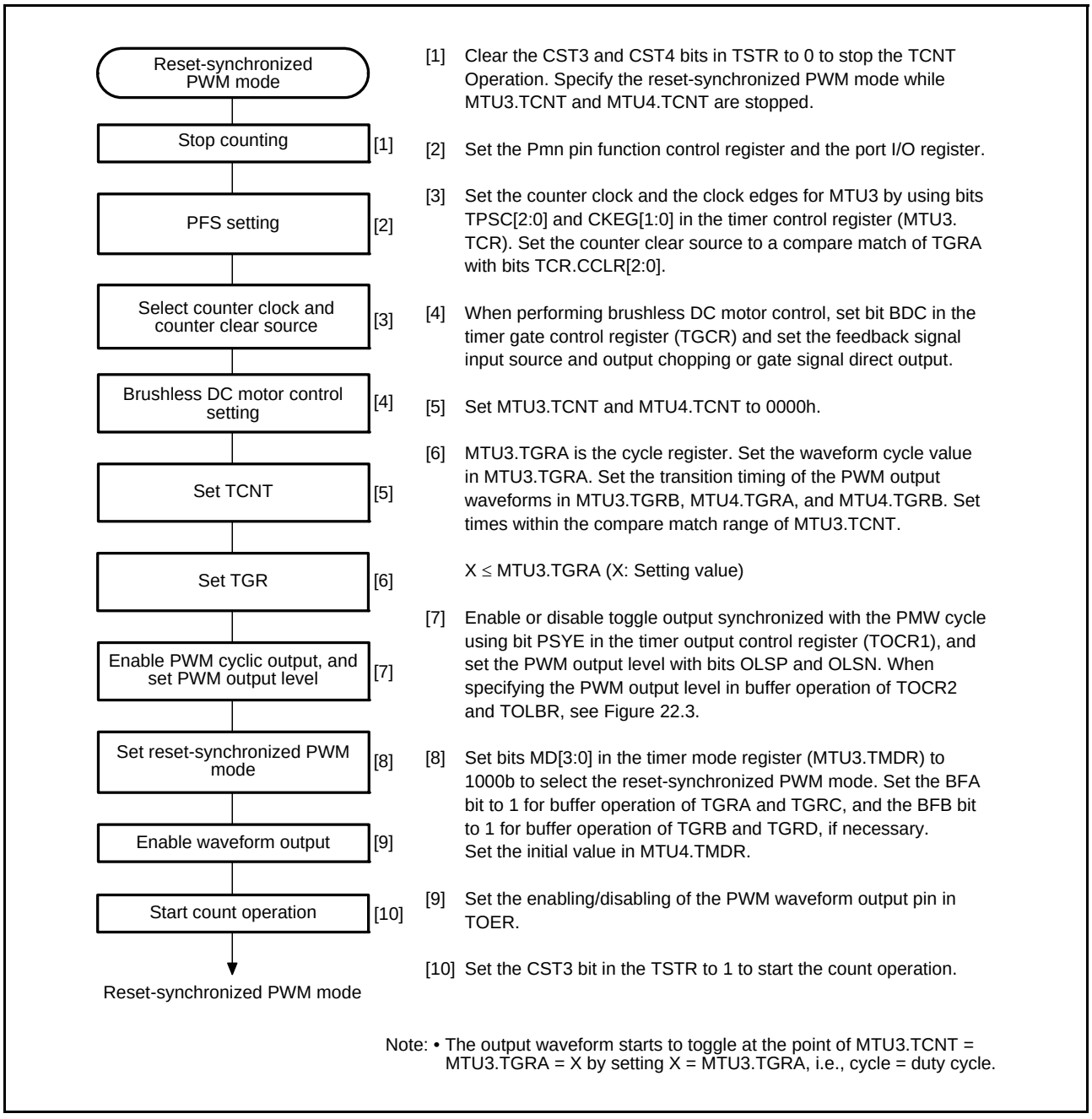


Figure 22.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 22.36 shows an example of operation in the reset-synchronized PWM mode.
MTU3.TCNT and MTU4.TCNT operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT and MTU3.TGRA, and then begin incrementing from 0000h. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB and the counters are cleared.

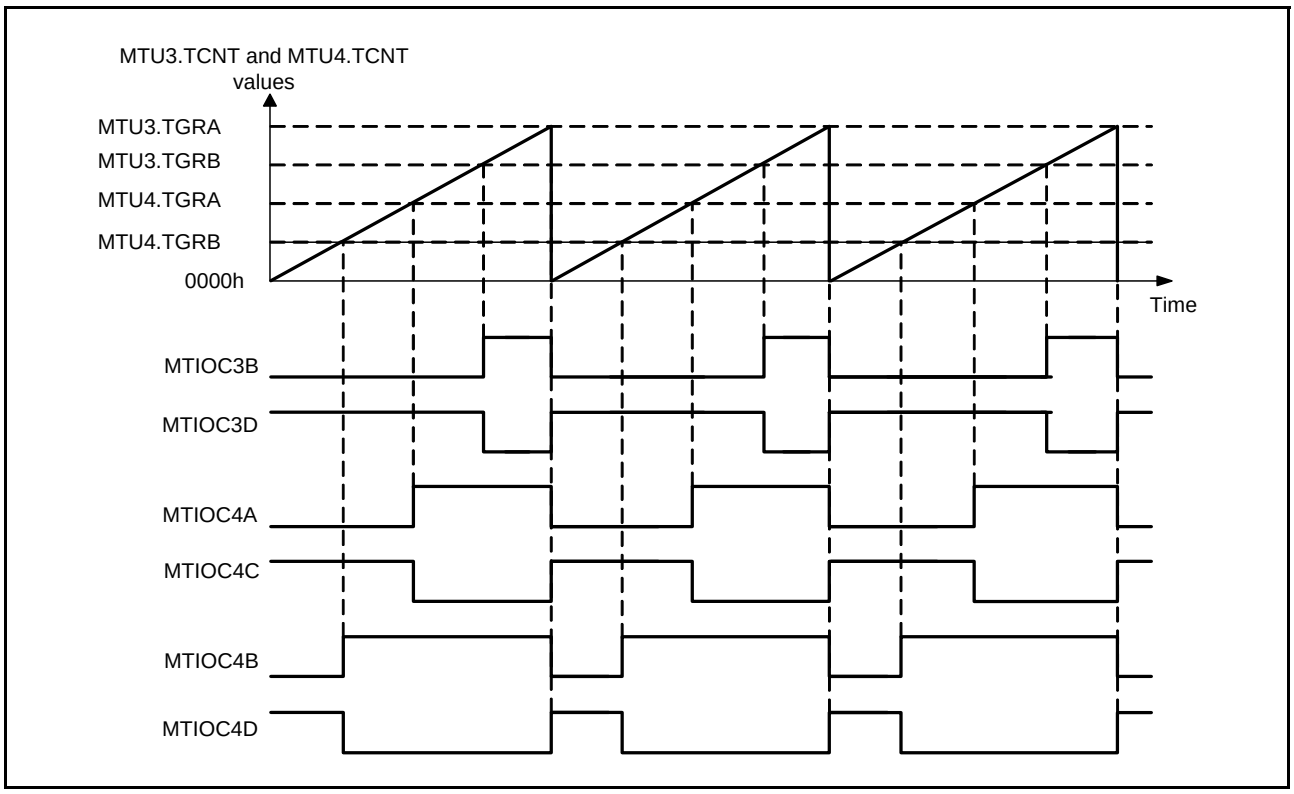


Figure 22.36 Example of Reset-Synchronized PWM Mode Operation (When TOCR's OLSN = 1 and OLSP = 1)

22.3.8 Complementary PWM Mode

In complementary PWM mode, three phases of non-overlapping positive and negative PWM waveforms can be output by combining MTU3 and MTU4. PWM waveforms without non-overlapping interval are also available.
In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins function as PWM output pins, and the MTIOC3A pin can be set for toggle output synchronized with the PWM cycle. MTU3.TCNT and MTU4.TCNT function as up/down-counters.
Table 22.54 lists the PWM output pins used. Table 22.55 lists 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 22.54 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' (non-overlapping negative-phase waveform of PWM output 1; PWM output without non-overlapping interval is also available)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2; PWM output without non-overlapping interval is also available)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3; PWM output without non-overlapping interval is also available)

Note 1. Avoid setting the MTIOC3C pin as a timer I/O pin in complementary PWM mode.

Table 22.55 Register Settings for Complementary PWM Mode

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3	MTU3.TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWER setting*1
	MTU3.TGRA	Set MTU3.TCNT upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting*1
	MTU3.TGRB	PWM output 1 compare register	Maskable by TRWER setting*1
	MTU3.TGRC	MTU3.TGRA buffer register	Always readable/writable
	MTU3.TGRD	PWM output 1/MTU3.TGRB buffer register	Always readable/writable
MTU4	MTU4.TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWER setting*1
	MTU4.TGRA	PWM output 2 compare register	Maskable by TRWER setting*1
	MTU4.TGRB	PWM output 3 compare register	Maskable by TRWER setting*1
	MTU4.TGRC	PWM output 2/MTU4.TGRA buffer register	Always readable/writable
	MTU4.TGRD	PWM output 3/MTU4.TGRB buffer register	Always readable/writable
Timer dead time data register (TDDR)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWER setting*1
Timer cycle data register (TCDR)		Set MTU4.TCNT upper limit value (1/2 carrier cycle)	Maskable by TRWER setting*1
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/MTU3.TGRB temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/MTU4.TGRA temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/MTU4.TGRB temporary register	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWER (timer read/write enable register).

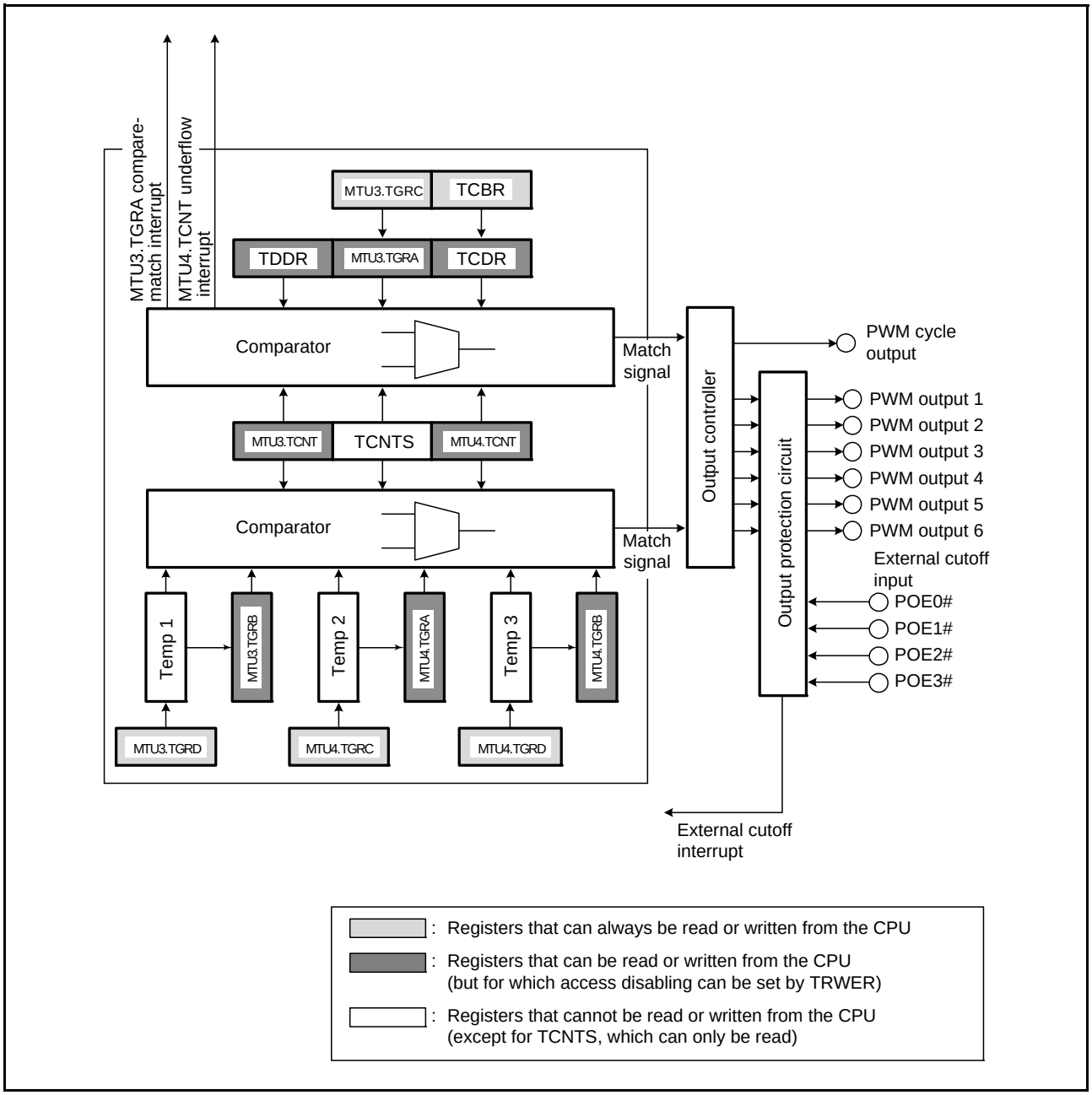
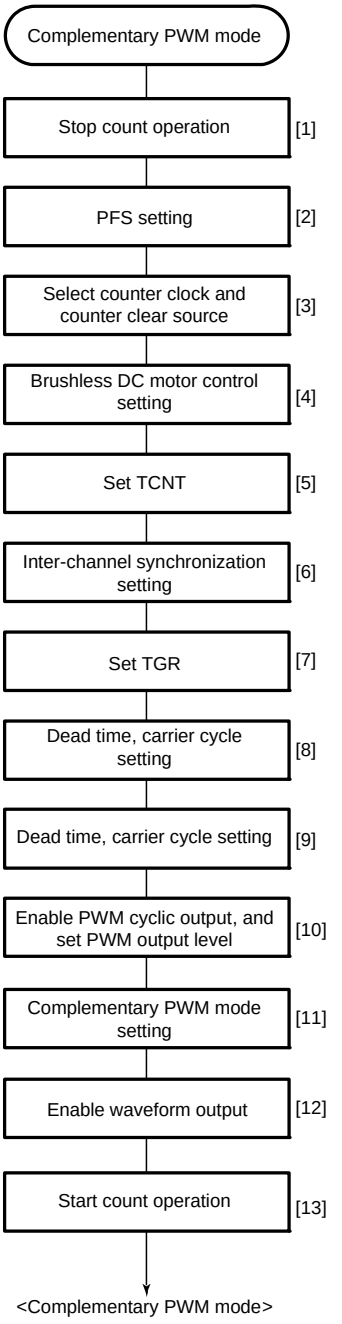


Figure 22.37 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 22.38 shows an example of the complementary PWM mode setting procedure.



[1] Clear bits CST3 and CST4 in the timer start register (TSTR) to 0 to stop timer counter (TCNT) operation. Specify complementary PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the Pmn pin function control register and the port I/O register.

[3] Set the same counter clock and the clock edges for MTU3 and MTU4 by using bits TPSC[2:0] and CKEG[1:0] in the timer control register (TCR).Set the synchronous clear for bits CCLR[2:0] only when restarting them by synchronized clearing from another channel during the complementary PWM mode operation.

[4] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[5] Set the dead time in MTU3.TCNT. Set MTU4.TCNT to 0000h.

[6] Set only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with MTU3 and MTU4 using the timer synchronous register (TSYR).

[7] Set the output PWM duty cycles in the duty registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD). Set the same value in the buffer registers and the duty registers.

[8] This setting is necessary only when no dead time should be generated. Make appropriate settings in the timer dead time enable register (TDER) so that no dead time is generated.

[9] Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the carrier cycle data register (TCDR) and carrier cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in MTU3.TGRA and MTU3.TGRC. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in MTU3.TGRA and MTU3.TGRC.

[10] Enable or disable toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1), and set the PWM output level with bits OLSP and OLSN.
When specifying the PWM output level by using TOLBR as a buffer for MTU2.TOCR, see Figure, Example of PWM Output Level Setting Procedure in Buffer Operation.

[11] Select complementary PWM mode in the timer mode register (MTU3.TMDR). Set the BFA bit to 1 for buffer operation of TGRA and TGRC, and the BFB bit to 1 for buffer operation of TGRB and TGRD, if necessary. Set nothing in MTU4.TMDR.

[12] Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).

[13] Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

Figure 22.38 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases of PWM waveforms can be output. Figure 22.39 illustrates counter operation in complementary PWM mode, and Figure 22.40 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTS— perform up-/down-count operations.

MTU3.TCNT is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

When the CST bit is set to 1, MTU3.TCNT counts up to the value set in MTU3.TGRA, then switches to down-counting when it matches MTU3.TGRA. When the MTU3.TCNT value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT should be initialized to 0000h.

When the CST bit is set to 1, MTU4.TCNT counts up in synchronization with MTU3.TCNT, and switches to down-counting when it matches TCDR. On reaching 0000h, MTU4.TCNT switches to up-counting, and the operation is repeated in this way.

TCNTS is a read-only counter. It does not need to be initialized.

When MTU3.TCNT matches TCDR during up-/down-counting of TCNT in MTU3 and MTU4, TCNTS starts down-counting, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches MTU3.TGRA, it is cleared to 0000h.

When MTU4.TCNT matches TDDR during down-counting of MTU3.TCNT and MTU4.TCNT, TCNTS starts up-counting, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches 0000h, it is set with the value in MTU3.TGRA.

TCNTS is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

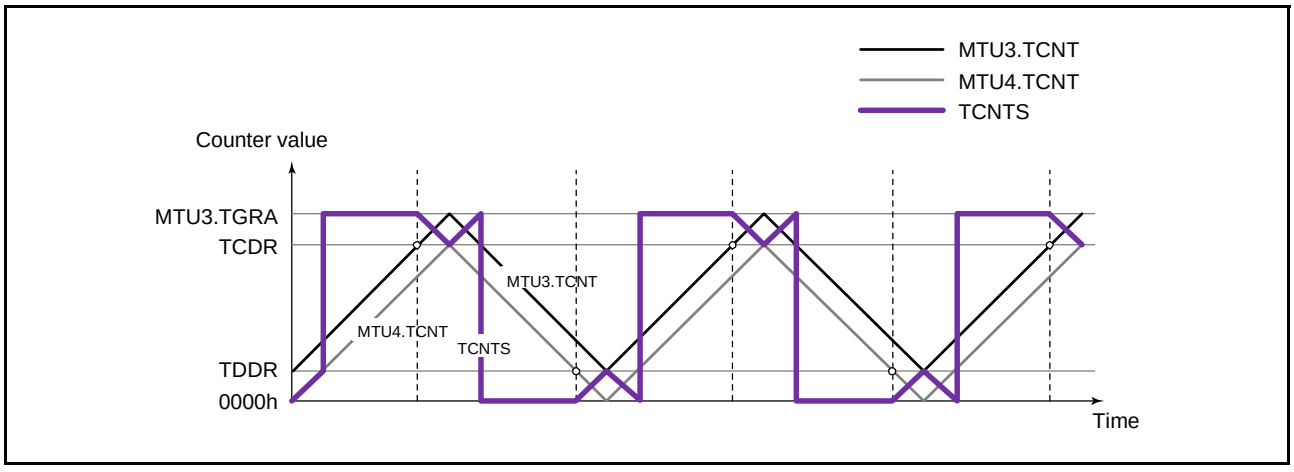


Figure 22.39 Counter Operation in Complementary PWM Mode

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used. Figure 22.40 shows an example of operation in complementary PWM mode.

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output.

MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD are buffer registers for these compare registers. Between a buffer register and a compare register, there is a temporary register. The temporary registers cannot be accessed by the CPU. 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.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches MTU3.TGRA while TCNTS is counting up, or 0000h while 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 (TMDR). Figure 22.40 shows an example in which the trough is selected for the transfer timing.

In the Tb (Tb1 in Figure 22.40) interval in which data is not transferred to the temporary register, 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 TCNTS) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

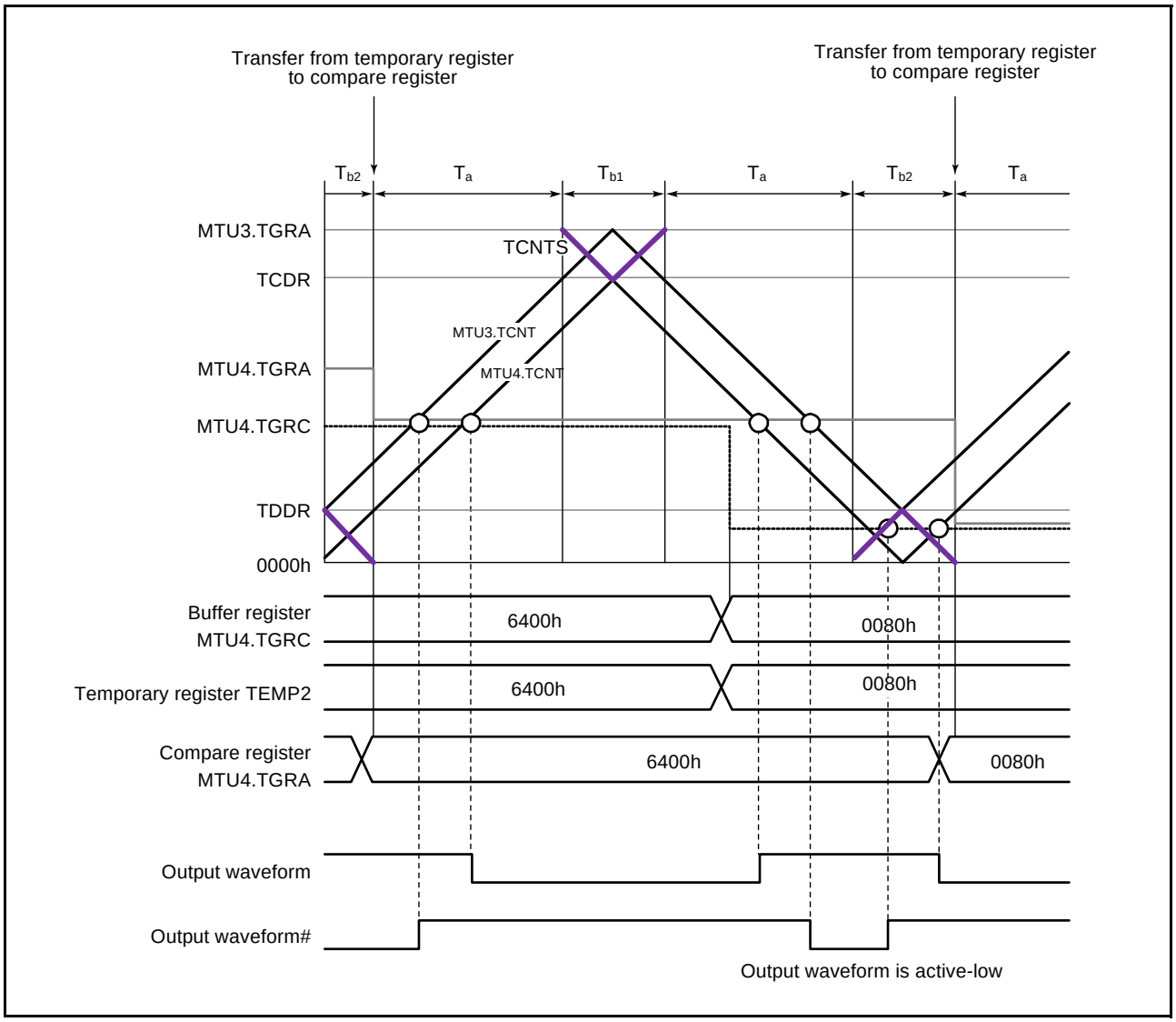


Figure 22.40 Example of Operation in Complementary PWM Mode

(c) Initial Setting

In complementary PWM mode, there are six 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 bits MD[3:0] in the timer mode register (TMDR), initial values should be set in the following registers.

MTU3.TGRC operates as the buffer register for MTU3.TGRA, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, MTU3.TGRC and MTU3.TGRA should be set to 1/2 the PWM carrier cycle + 1, and TDDR should be set to 1.

Set the respective initial PWM duty values in three buffer registers MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD.

The values set in the five buffer registers excluding TDDR are transferred to the corresponding compare registers as soon as complementary PWM mode is set.

Set MTU4.TCNT to 0000h before setting complementary PWM mode.

Table 22.56 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
MTU3.TGRC	1/2 PWM carrier cycle + dead time Td (1/2 PWM carrier cycle + 1 when dead time generation is disabled by TDER)
TDDR	Dead time Td (1 when dead time generation is disabled by TDER)
TCBR	1/2 PWM carrier cycle
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD	Initial PWM duty value for each phase
MTU4.TCNT	0000h

Note: • The value set in MTU3.TGRC should be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR. When dead time generation is disabled by TDER, TGRC should be set to 1/2 the PWM carrier cycle + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2).
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, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.
The non-overlap time is set in the timer dead time data register (TDDR). The value set in TDDR is used as the MTU3.TCNT counter start value and creates a non-overlapping interval between MTU3.TCNT and MTU4.TCNT.
Complementary PWM mode should be cleared before changing the contents of TDDR.

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER bit can be cleared to 0 only when 0 is written to it after reading TDER = 1.
MTU3.TGRA and MTU3.TGRC should be set to 1/2 PWM carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 22.41 shows an example of operation without dead time.

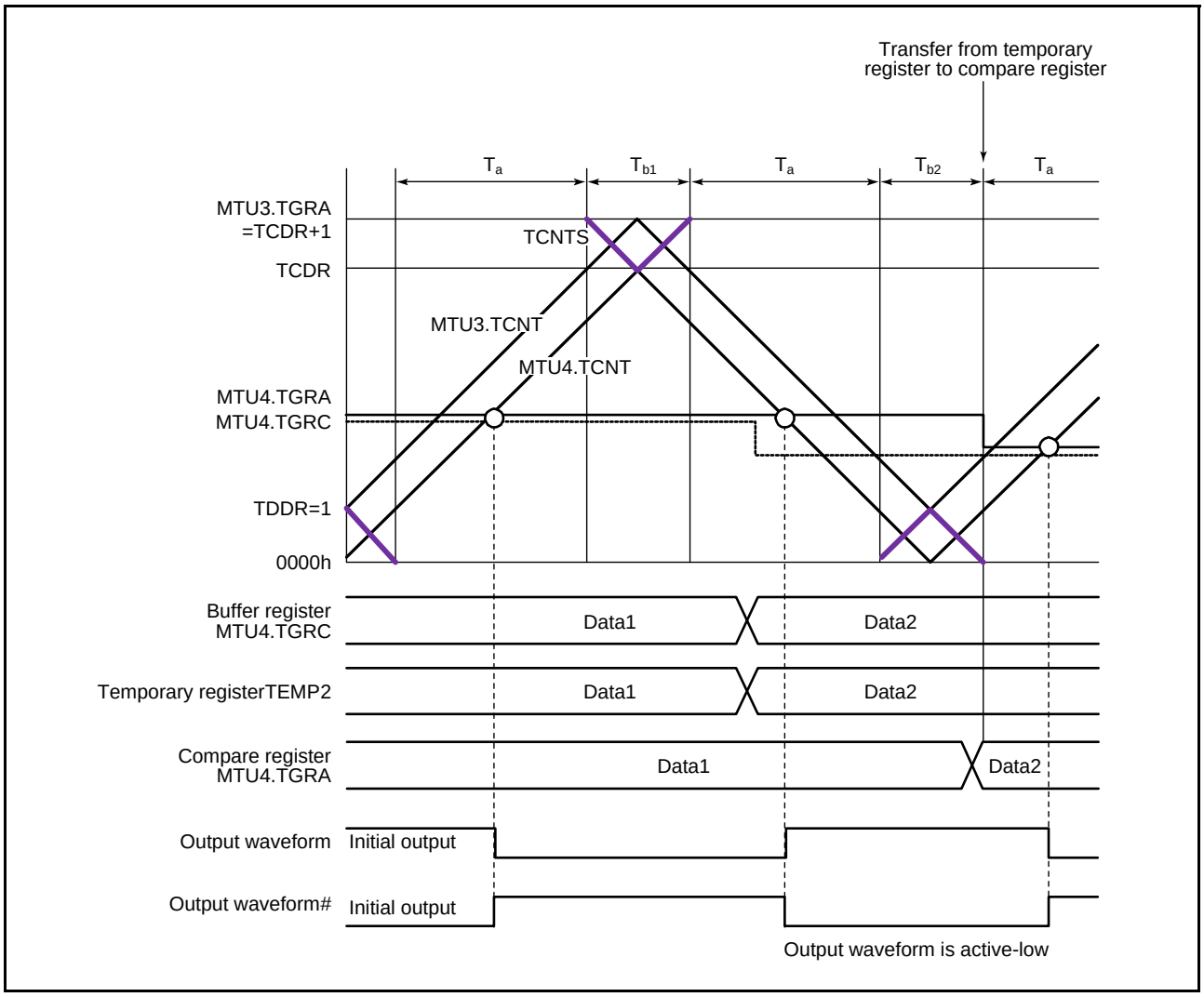


Figure 22.41 Example of Operation without Dead Time

(g) PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—MTU3.TGRA, in which the MTU3.TCNT upper limit value is set, and TCDR, in which the MTU4.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 setting = TCDR setting + TDDR setting
Without dead time: MTU3.TGRA setting = TCDR setting + 1

The MTU3.TGRA and TCDR settings are made by setting values in buffer registers MTU3.TGRC and TCBR. The values set in MTU3.TGRC and TCBR are transferred simultaneously to MTU3.TGRA and TCDR with the transfer timing selected with bits MD[3:0] in the timer mode register (TMDR).
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 22.42 illustrates the operation when the PWM cycle is updated at the crest.
See the following section (h), Register Data Updating, for the method of updating the data in each buffer register.

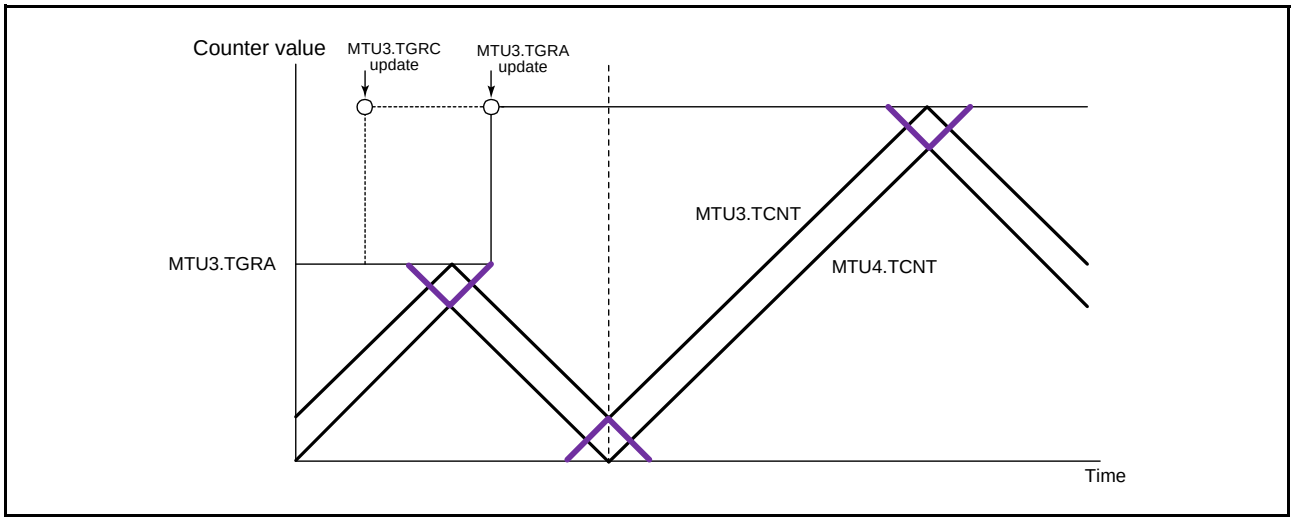


Figure 22.42 Example of PWM Cycle Updating

(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 carrier 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 TCNTS 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 TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD[3:0] in the timer mode register (TMDR). Figure 22.43 shows an example of data updating in complementary PWM mode. 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 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.

Even when not updating all five registers or when not updating the MTU4.TGRD data, be sure to write to MTU4.TGRD after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD should be the same as the data prior to the write operation.

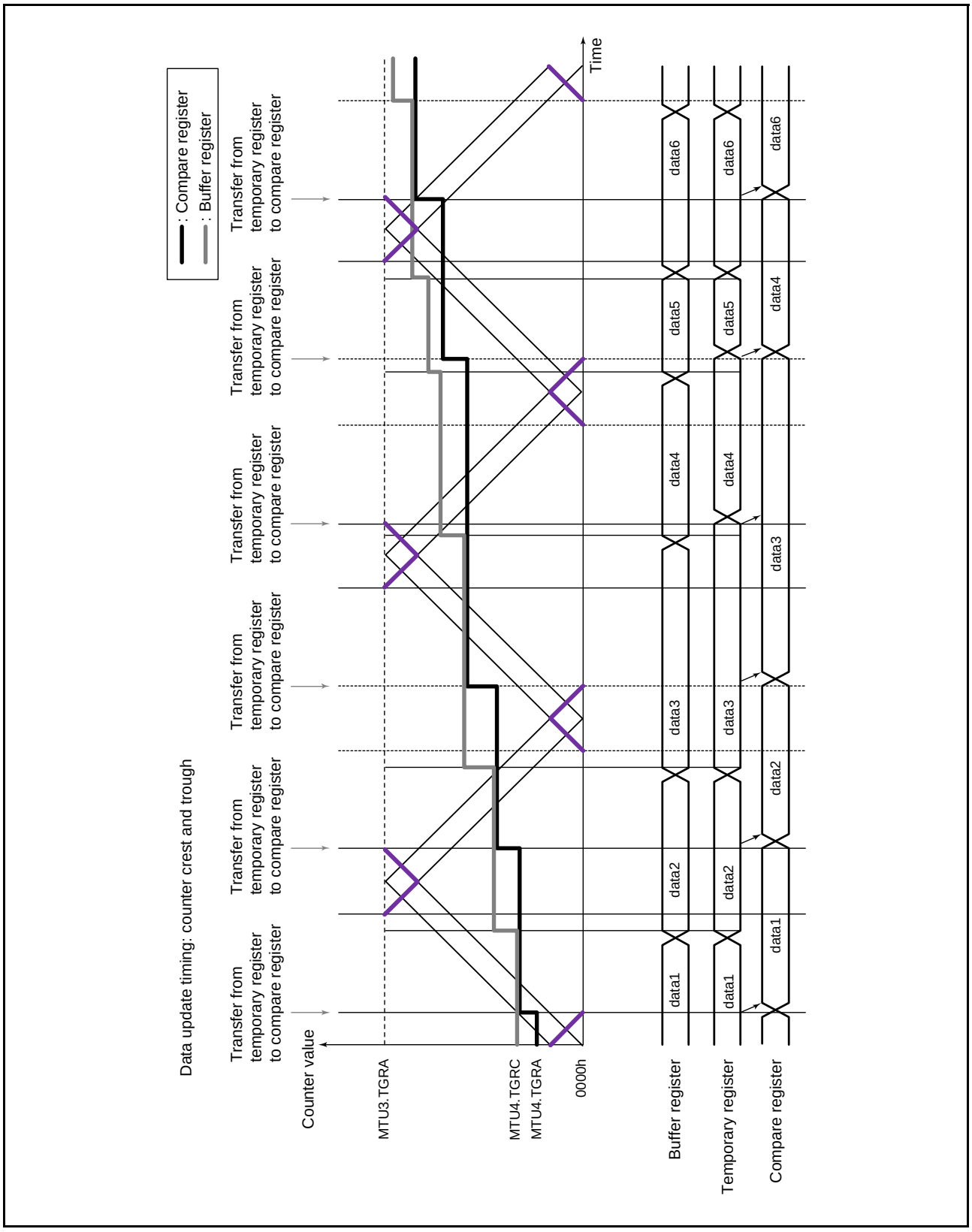


Figure 22.43 Example of Data Updating in Complementary PWM Mode

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2). This initial output is the non-active level of the PWM pulse and continues from when complementary PWM mode is set with the timer mode register (TMDR) until MTU4.TCNT exceeds the value set in the dead time register (TDDR). Figure 22.44 shows an example of the initial output in complementary PWM mode.
An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in Figure 22.45.

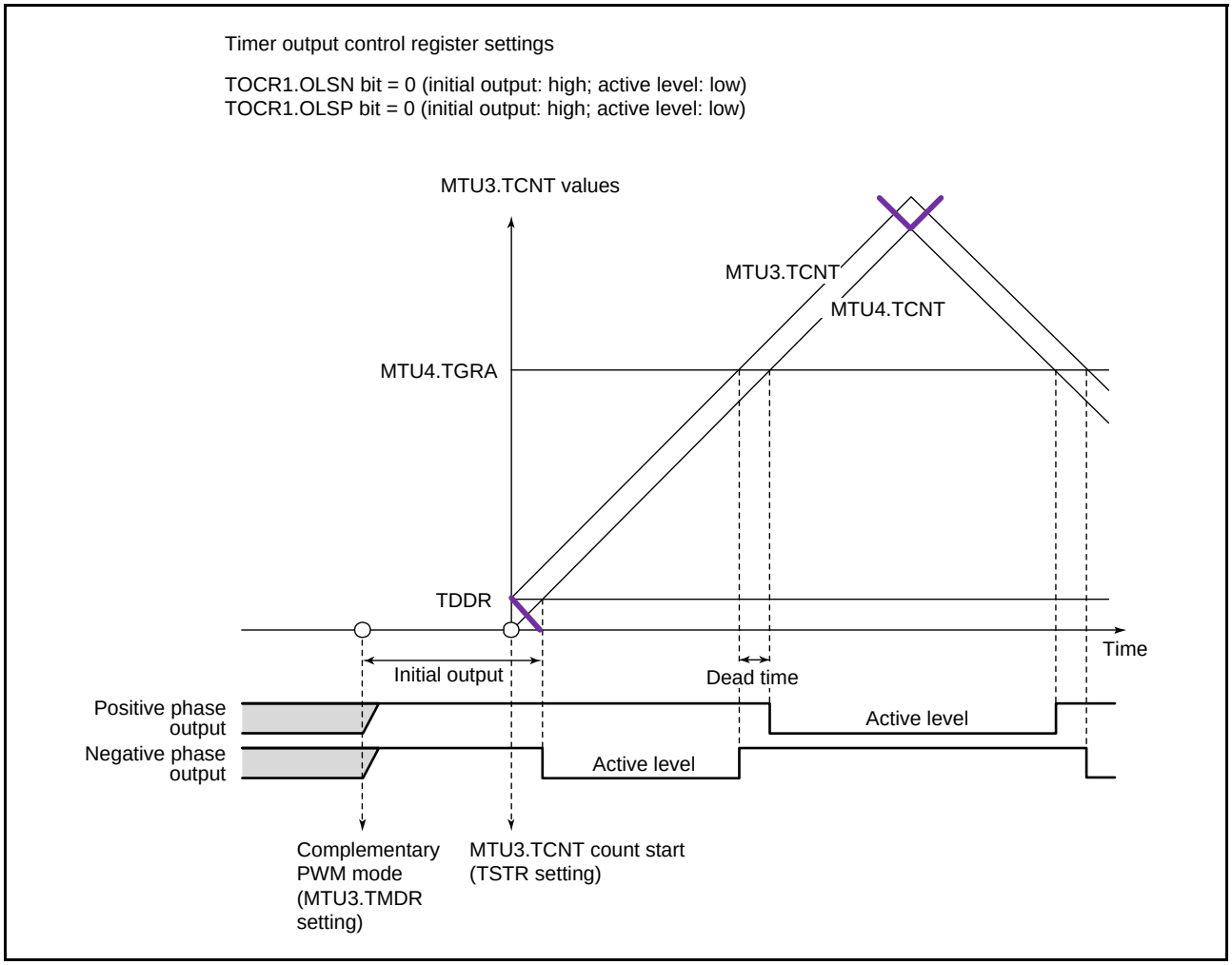


Figure 22.44 Example of Initial Output in Complementary PWM Mode (1)

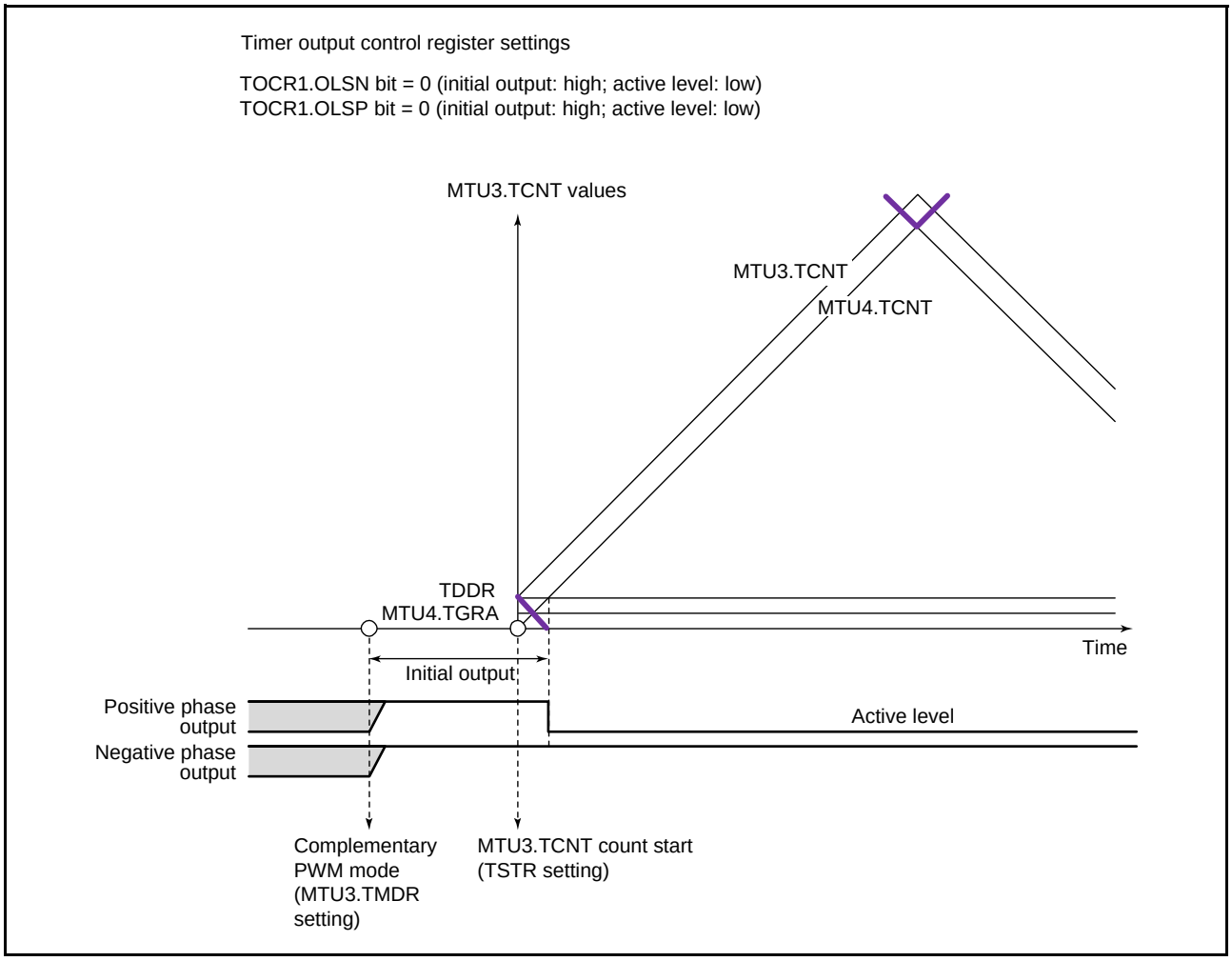


Figure 22.45 Example of Initial Output in Complementary PWM Mode (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, three phases of PWM waveforms are output with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.
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 data register. While TCNTS is counting, the data register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. 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 22.46 to Figure 22.48 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 value indicated in solid lines in the figure, and the turn-on timing by a compare match with the counter indicated in dotted lines, which operates with a delay of the dead time behind the solid-line counter. 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 22.46.
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 22.47, 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 higher 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 22.48, compare match a' with new data in the temporary register occurs before compare match c, but until compare match c, which turns off the positive phase, other compare matches are ignored. 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.

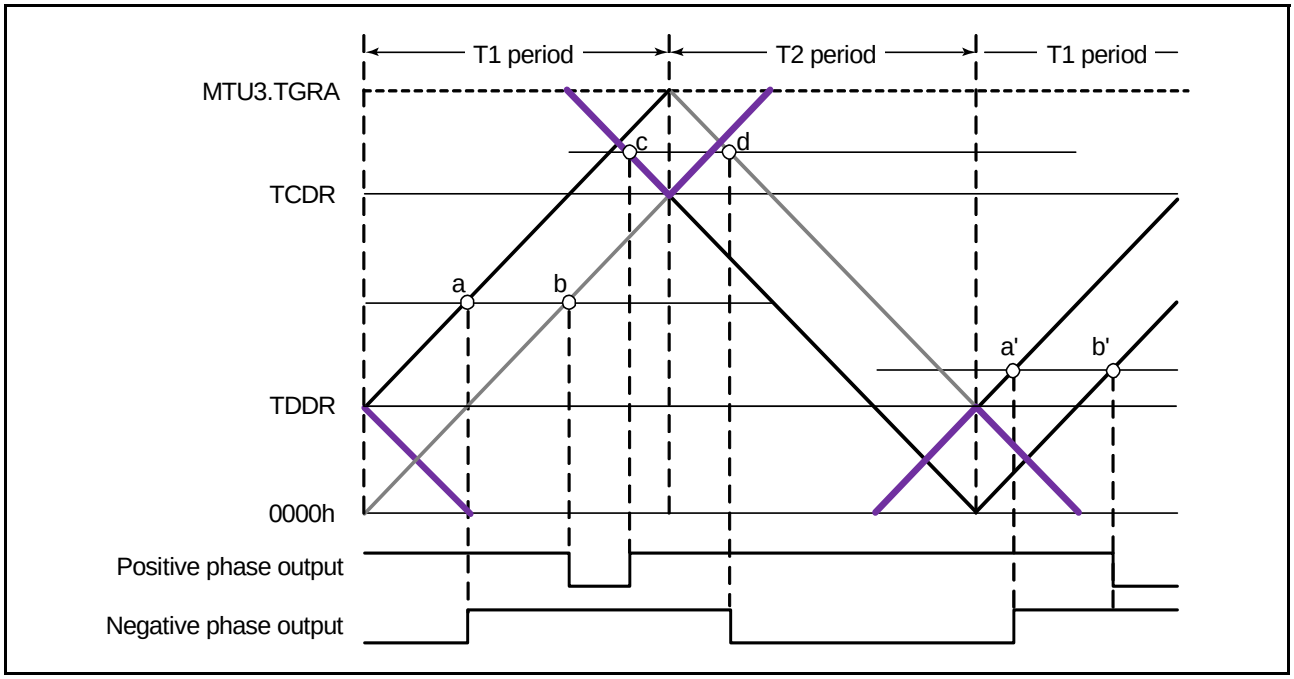


Figure 22.46 Example of Waveform Output in Complementary PWM Mode (1)

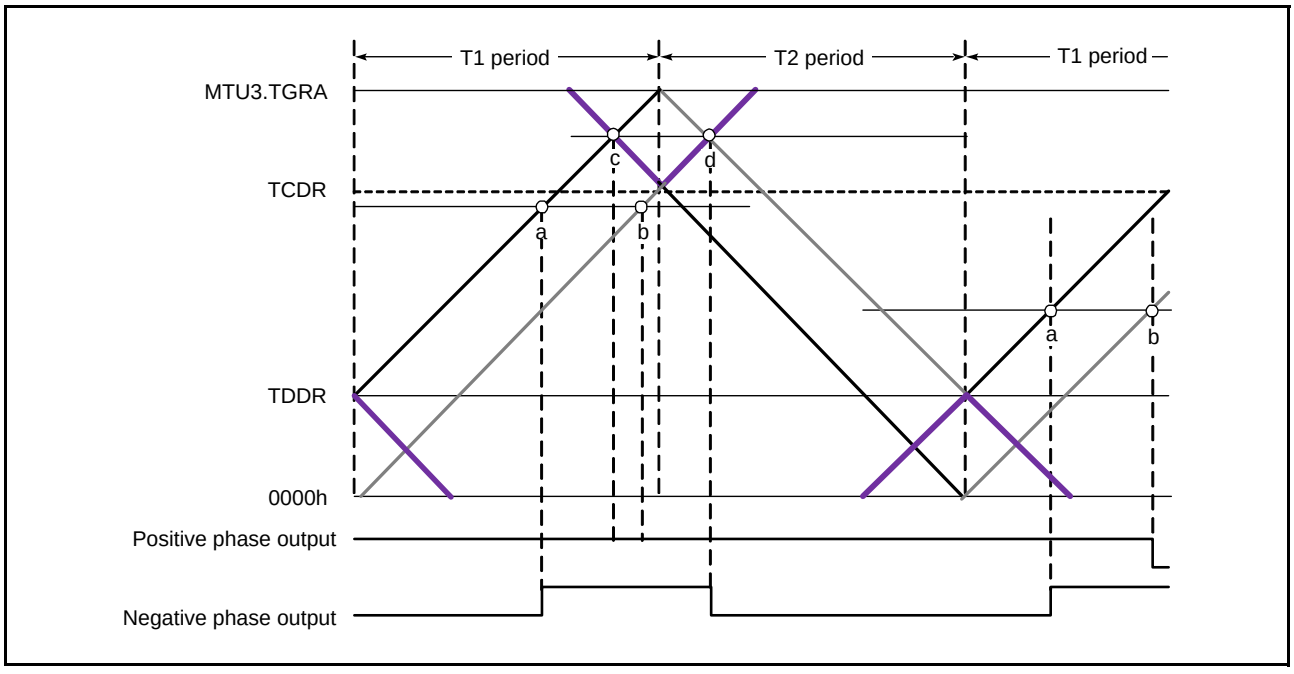


Figure 22.47 Example of Waveform Output in Complementary PWM Mode (2)

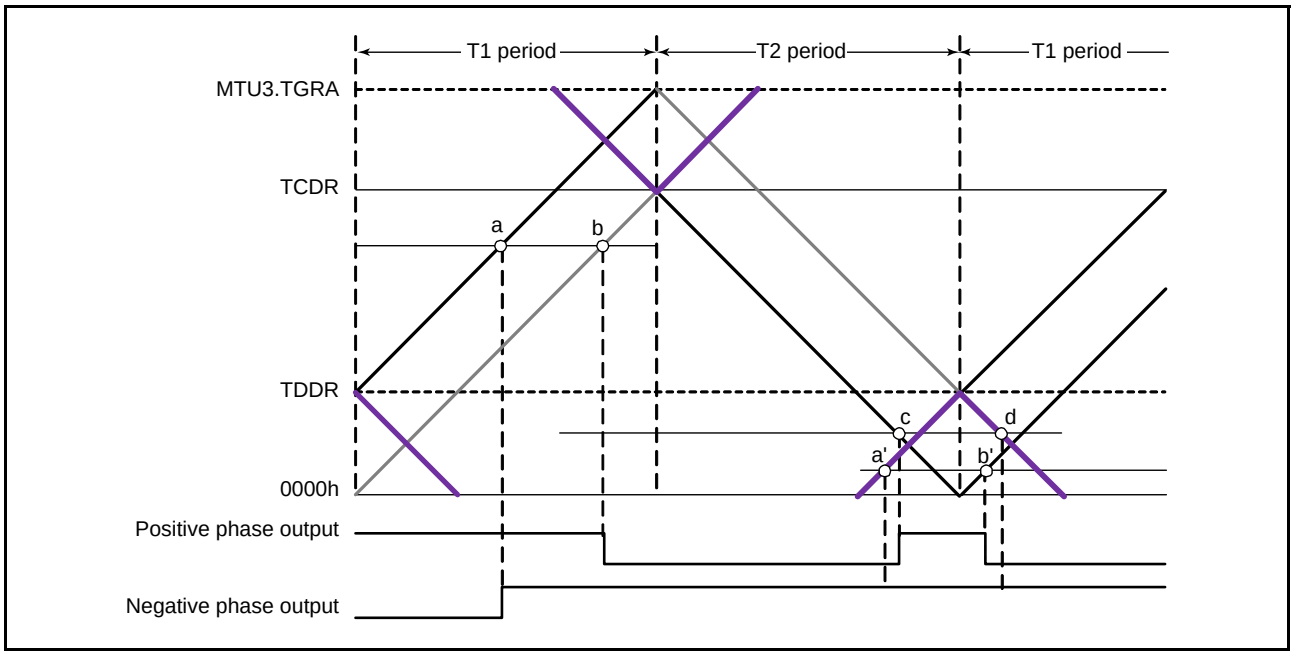


Figure 22.48 Example of Waveform Output in Complementary PWM Mode (3)

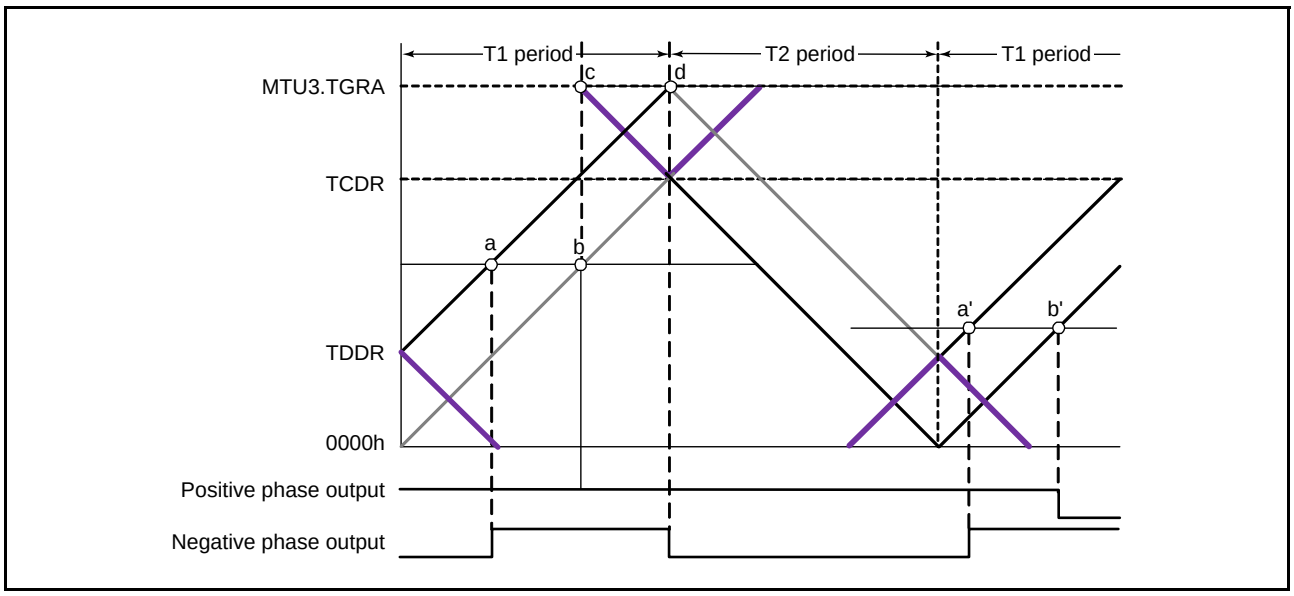


Figure 22.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1)

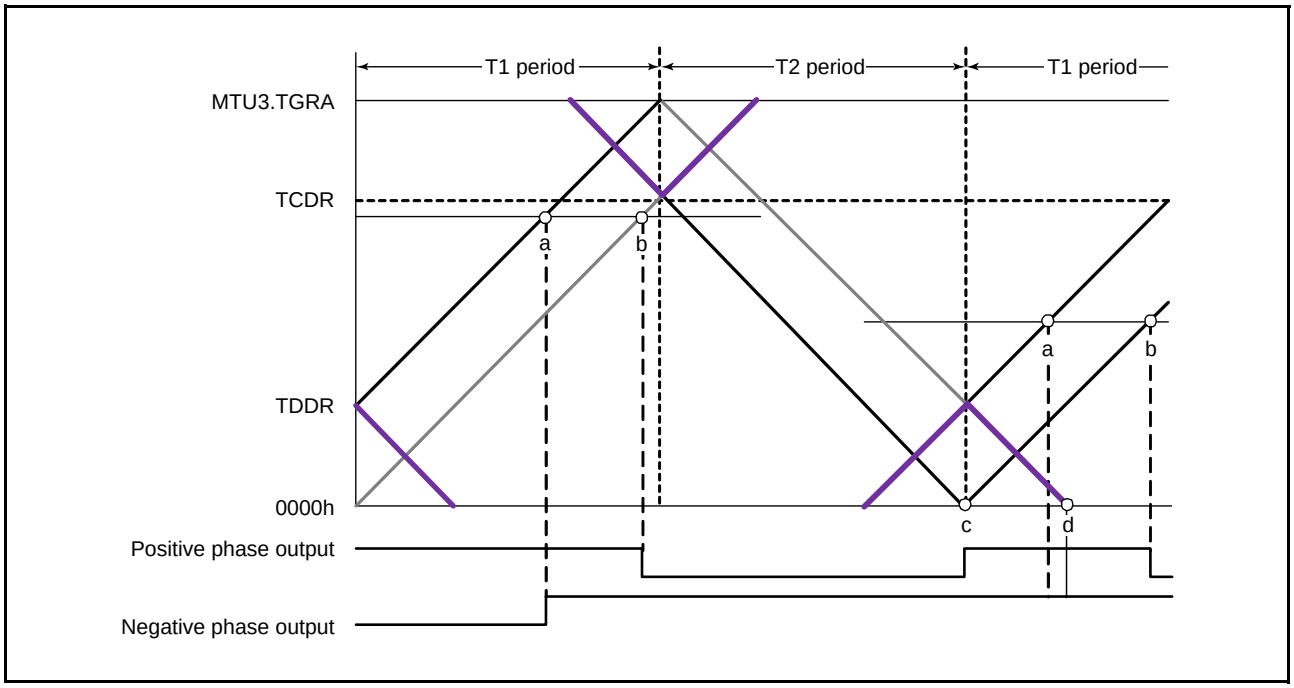


Figure 22.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2)

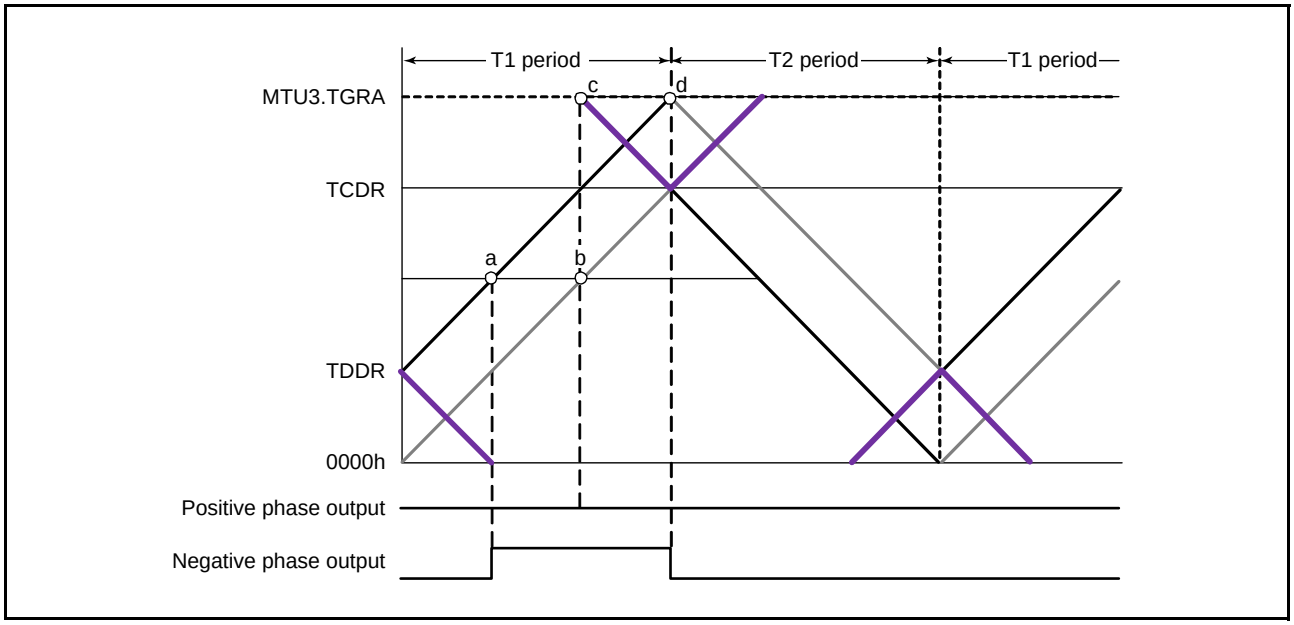


Figure 22.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3)

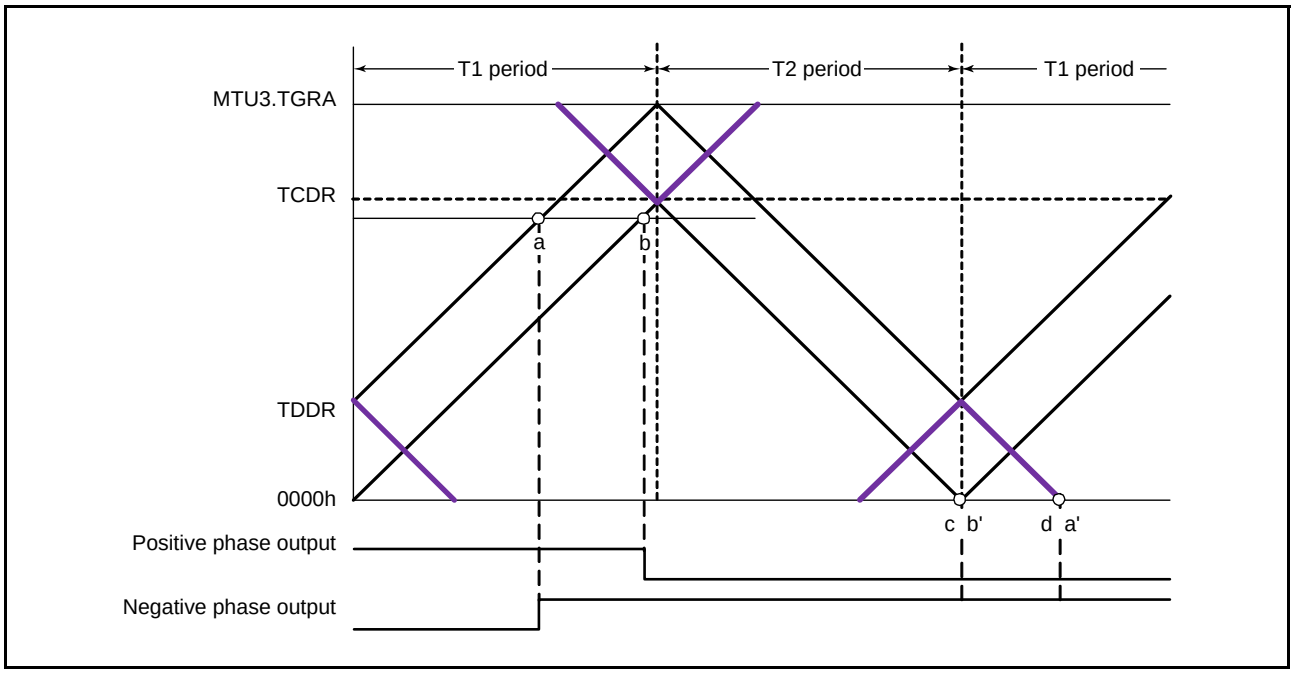


Figure 22.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4)

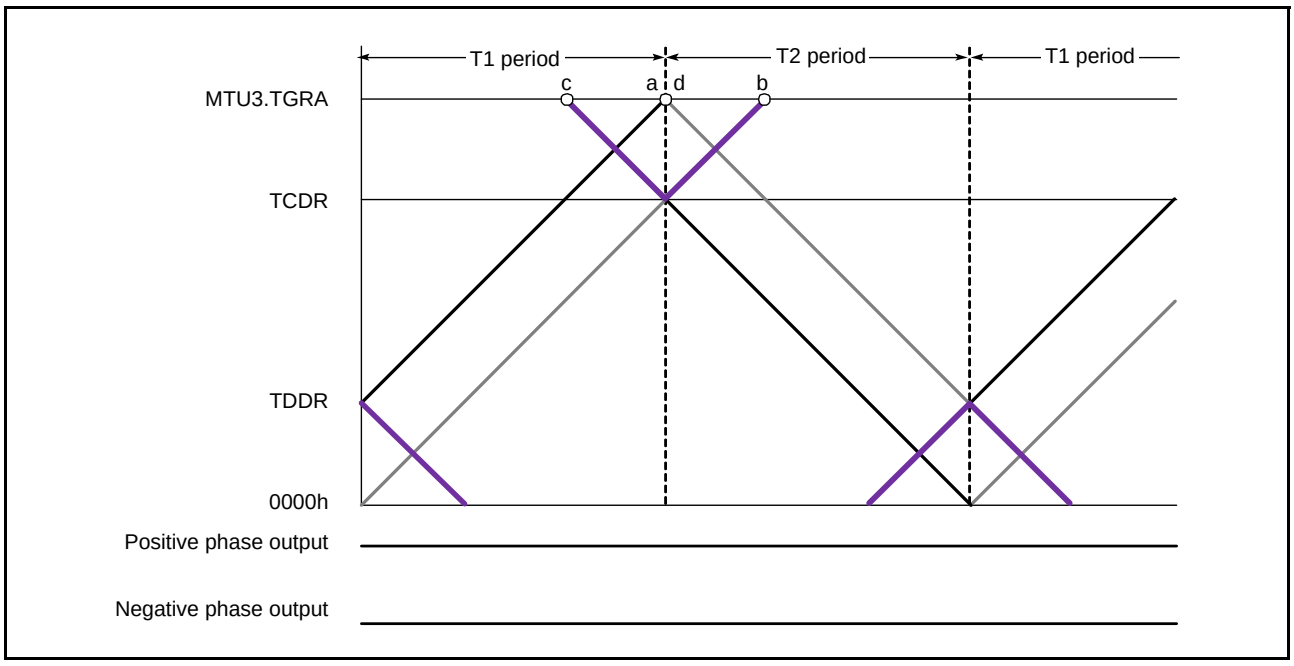


Figure 22.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5)

(k) 0% and 100% Duty Cycle Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty cycle PWM waveforms can be output as required. Figure 22.49 to Figure 22.52 show output examples.

A 100% duty cycle waveform is output when the data register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty cycle waveform is output when the data register value is set to the same value as MTU3.TGRA. The waveform in this case has a positive phase with a 100% off-state.

On and 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.

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output in synchronization with the PWM carrier cycle can be generated by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in Figure 22.54.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA and a compare match between MTU4.TCNT and 0000h.

The MTIOC3A pin is assigned for this toggle output. The initial output is 1.

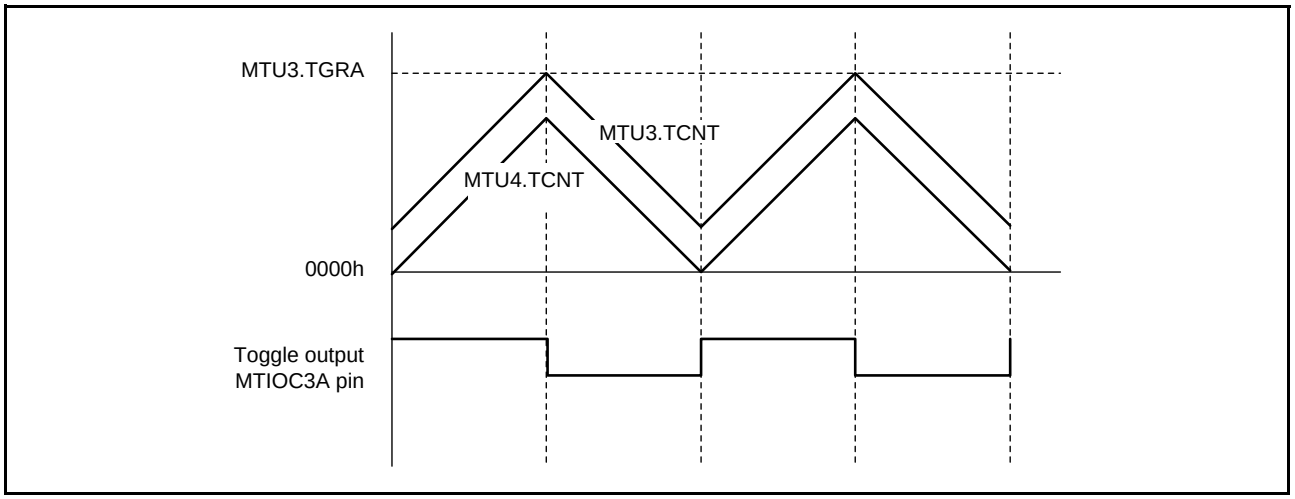


Figure 22.54 Example of Toggle Output Waveform Synchronized with PWM Output

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by another channel when a mode for synchronization with another channel is specified through the timer synchronous register (TSYR) and synchronous clearing is selected with bits CCLR[2:0] in the timer control register (TCR).

Figure 22.55 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

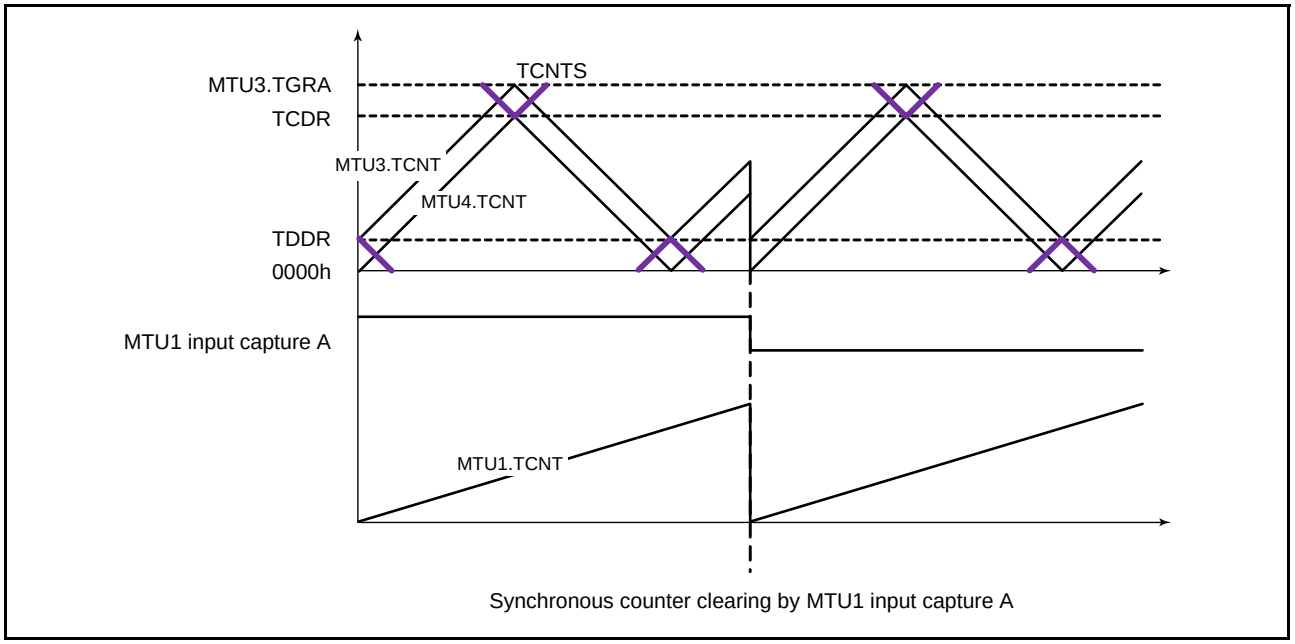


Figure 22.55 Counter Clearing Synchronized with Another Channel

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing. Initial output suppression through setting WRE bit to 1 is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in Figure 22.56. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial output period (indicated by (1) in Figure 22.56) immediately after the counters start operation, initial value output is not suppressed.
Synchronous clearing generated in MTU0 to MTU2 can cause counter clearing in the MTU.

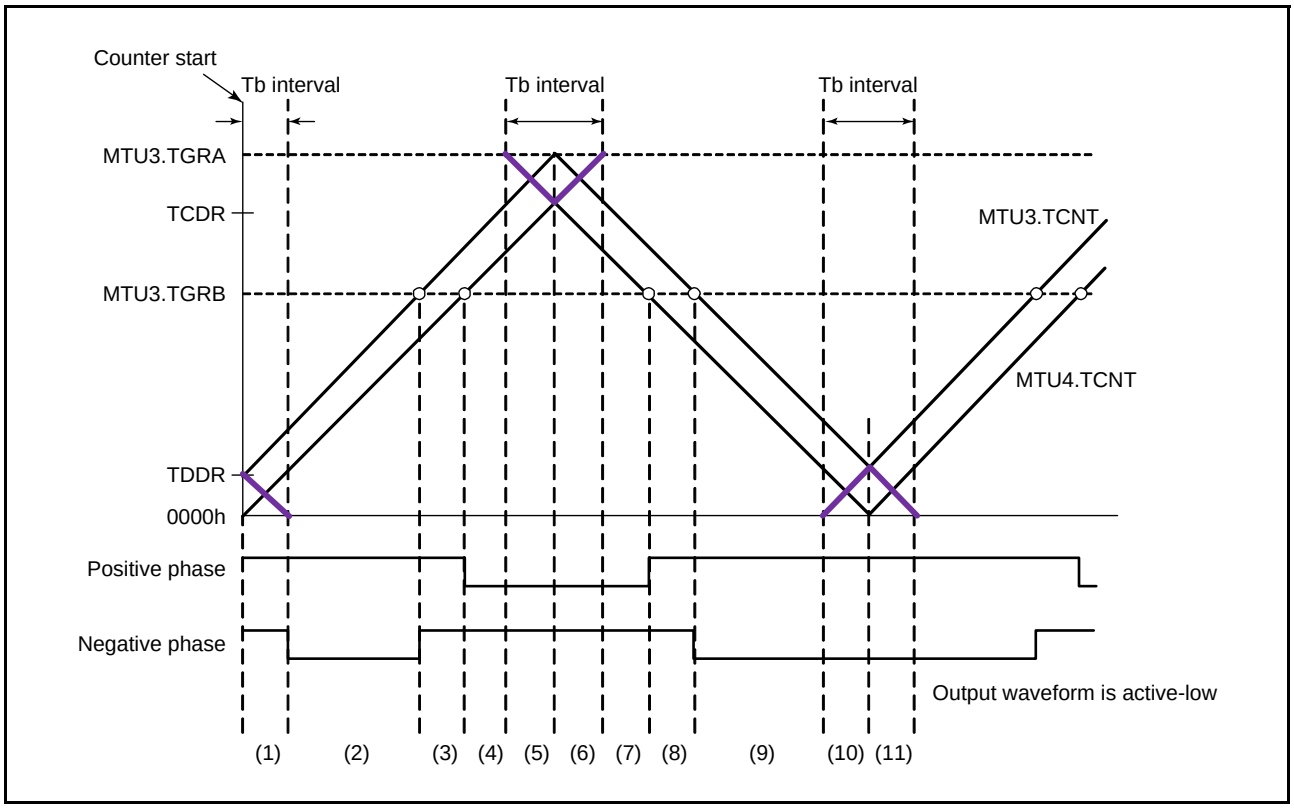


Figure 22.56 Timing for Synchronous Counter Clearing

- 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 22.57.

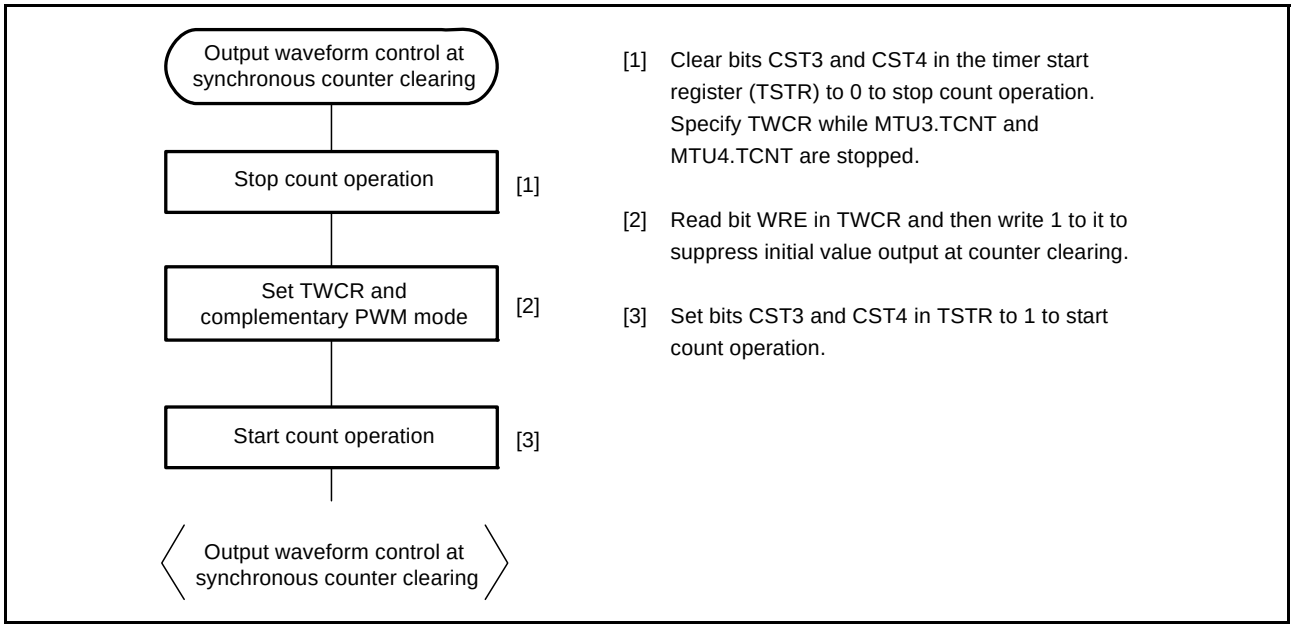


Figure 22.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 22.58 to Figure 22.61 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCR is set to 1. In the examples shown in Figure 22.58 to Figure 22.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 22.56, respectively.

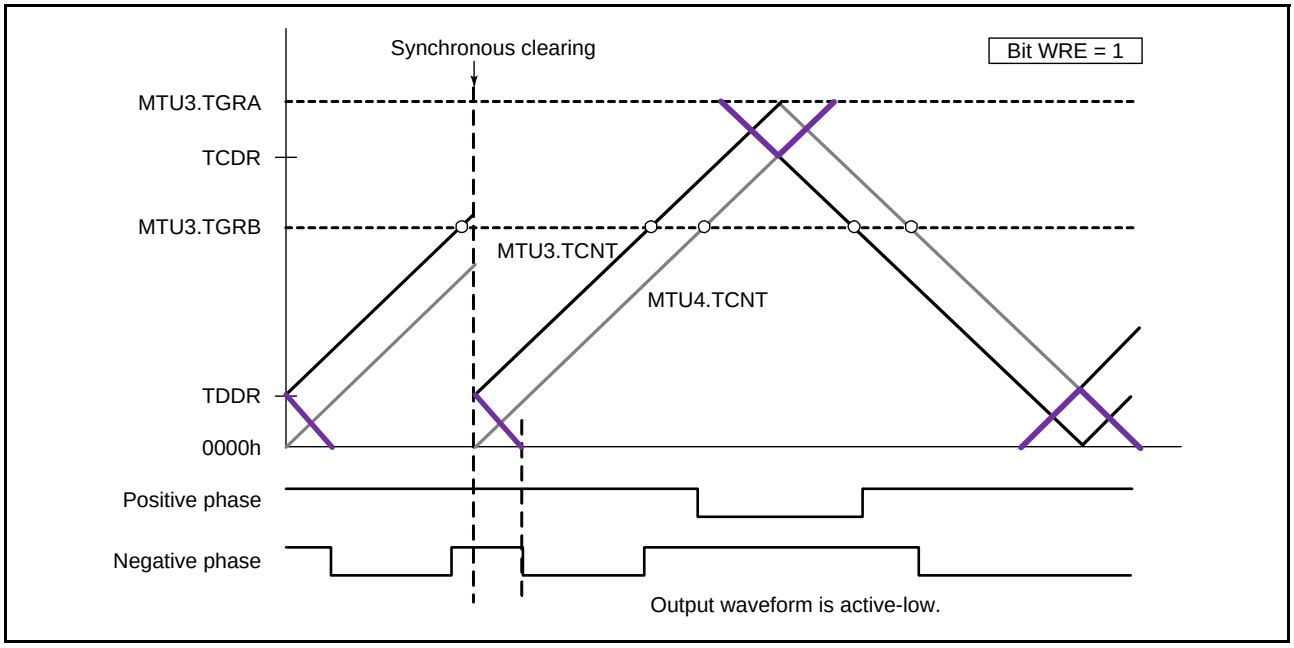


Figure 22.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 22.56; Bit WRE of TWCR is 1 in the MTU)

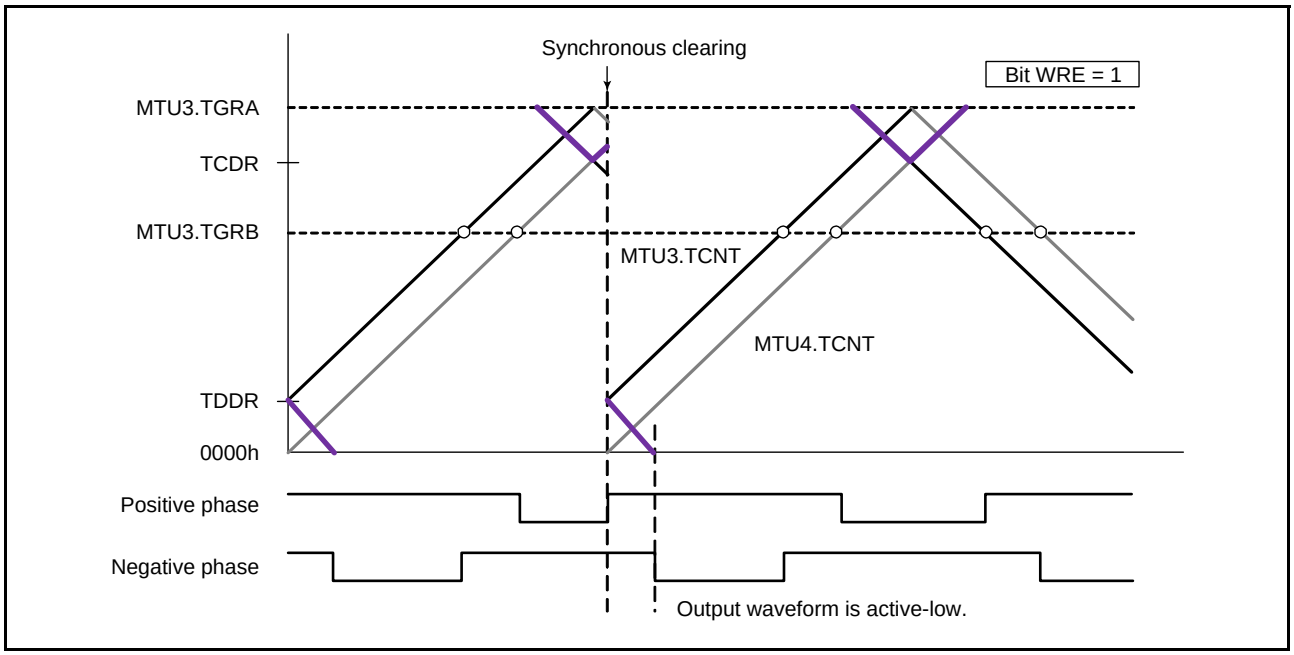


Figure 22.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 22.56; Bit WRE of TWCR is 1 in the MTU)

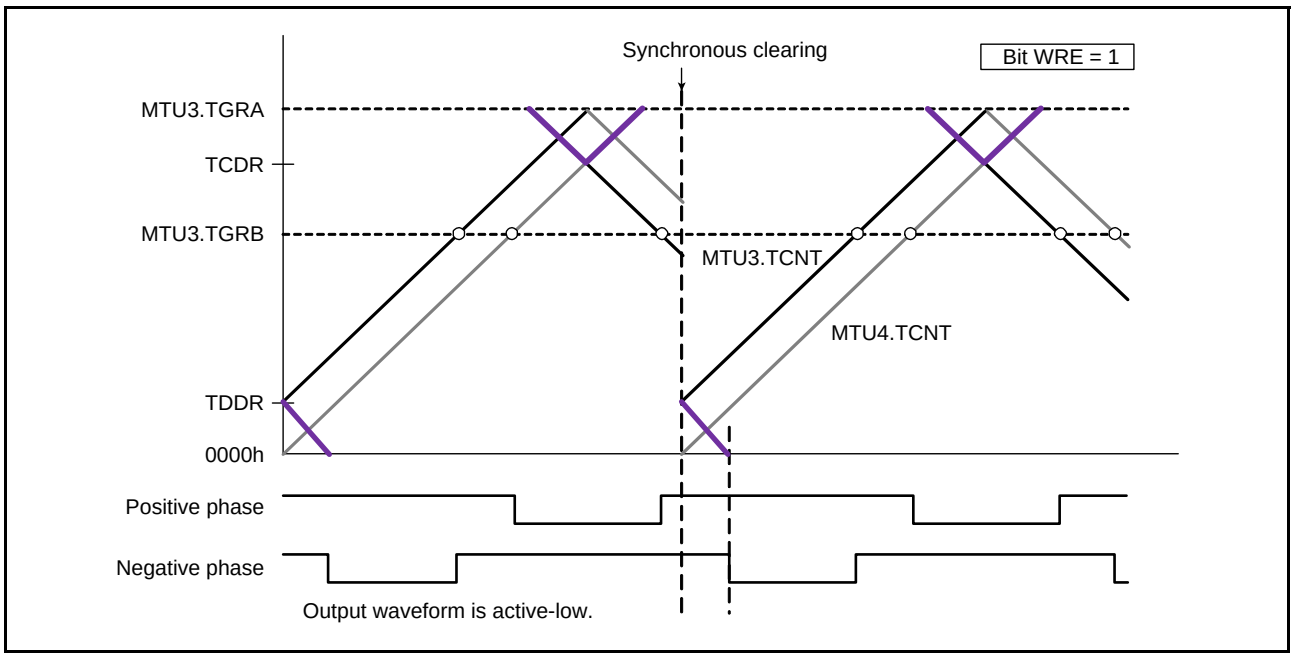


Figure 22.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 22.56; Bit WRE of TWCR is 1)

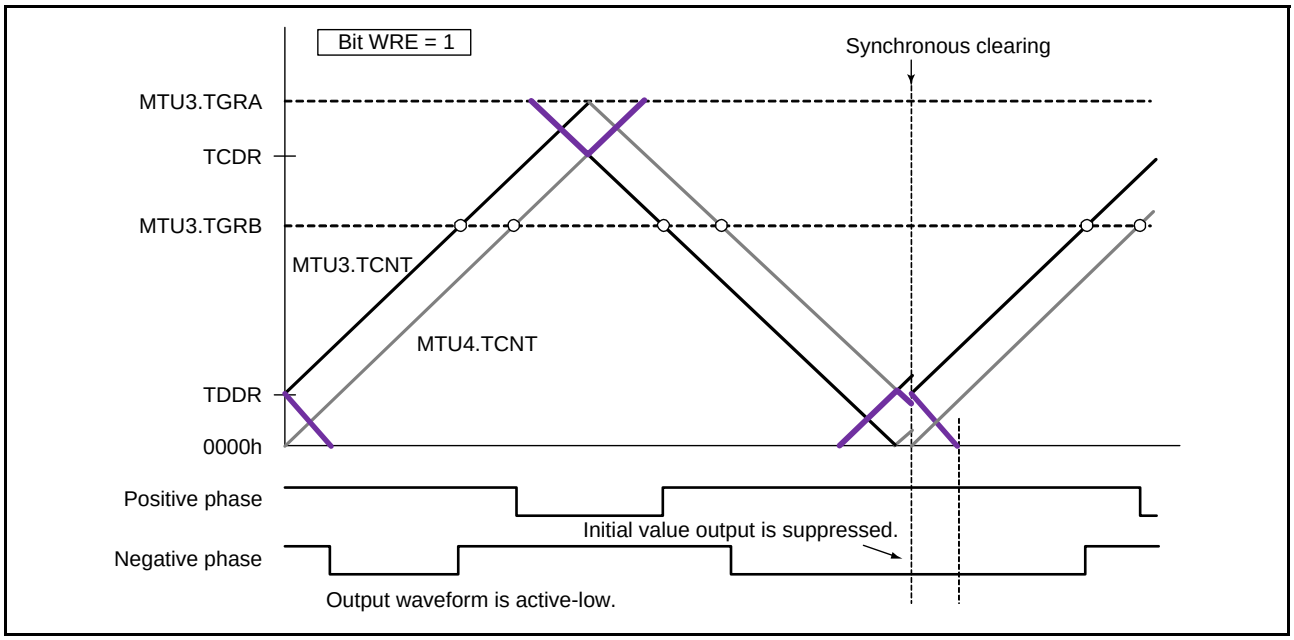


Figure 22.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 22.56; Bit WRE of TWCR is 1)

(o) Counter Clearing by MTU3.TGRA Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by MTU3.TGRA compare match when the CCE bit is set in the timer waveform control register (TWCR).

Figure 22.62 illustrates an operation example.

Note: • Use this function only in complementary PWM mode 1 (transfer at crest).
Note: • Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1).
Note: • Do not set the PWM duty cycle value to 0000h.
Note: • Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

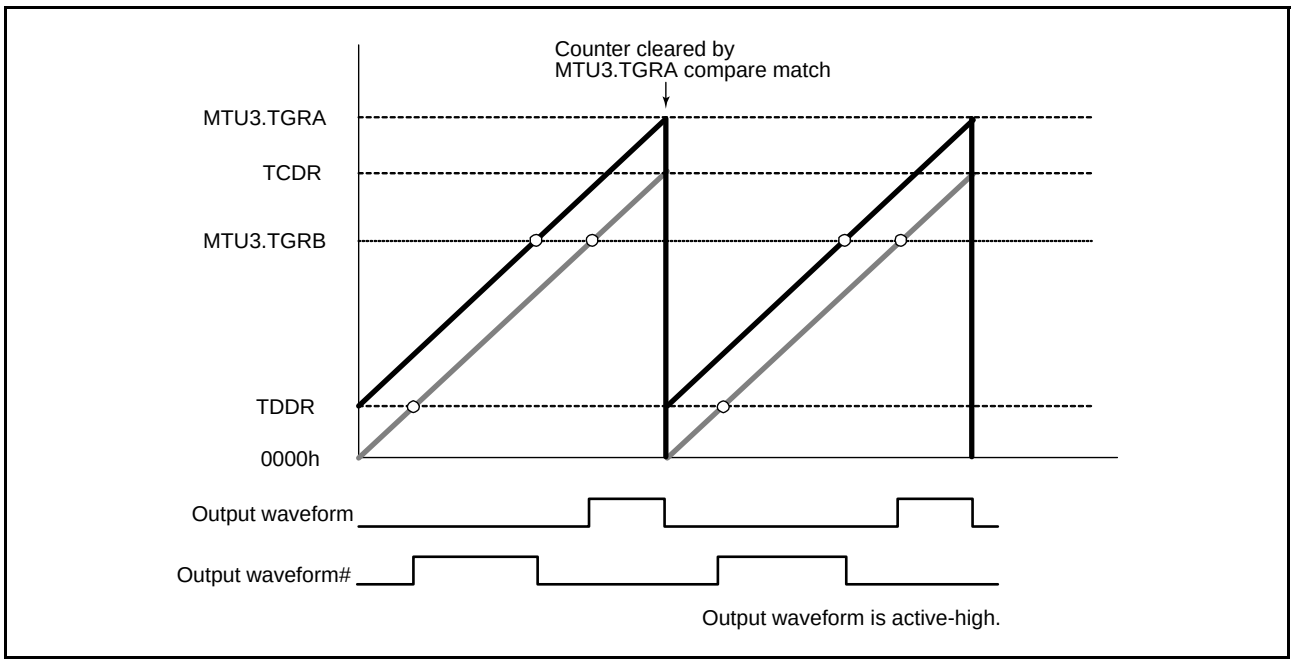


Figure 22.62 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(p) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figure 22.63 to Figure 22.66 show examples of brushless DC motor driving waveforms created using TGCR.
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 FB bit in TGCR 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 (set with PFS). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.
When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1.
The driving waveforms are output from the 6-phase 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 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 OLSN and OLSP bits in the timer output control register (TOCR) regardless of the setting of the N and P bits.

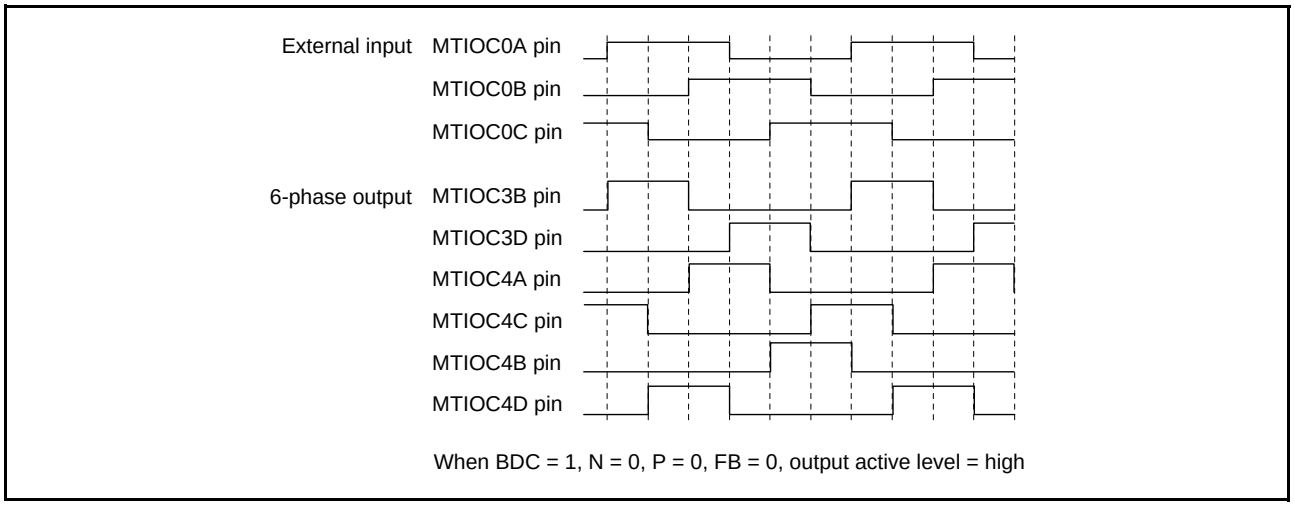


Figure 22.63 Example of Output Phase Switching by External Input (1)

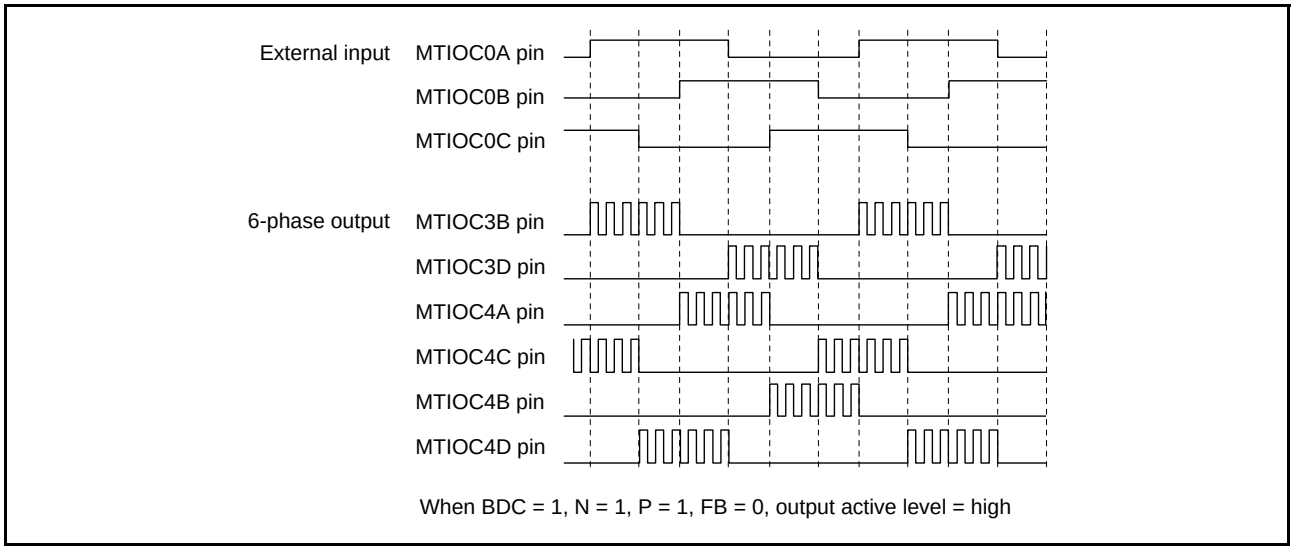


Figure 22.64 Example of Output Phase Switching by External Input (2)

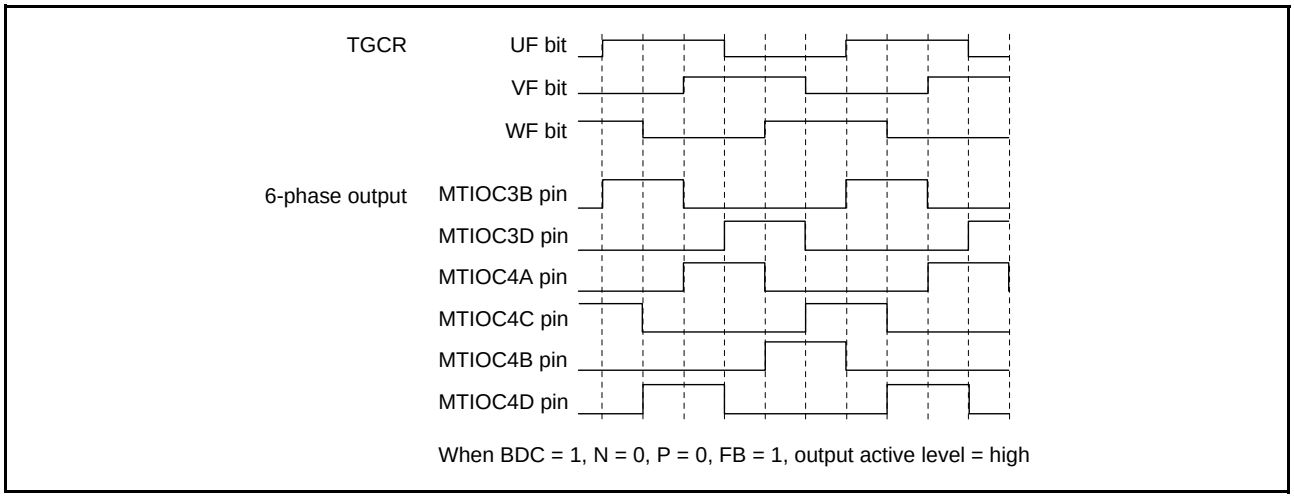


Figure 22.65 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)

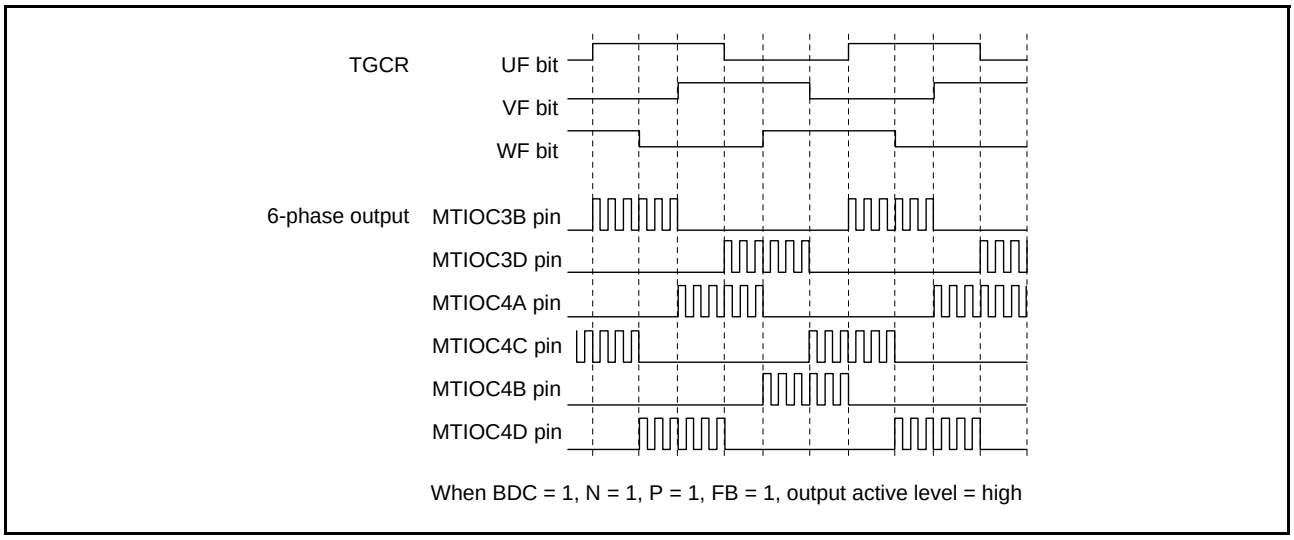


Figure 22.66 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTU3.TGRA compare match, MTU4.TCNT underflow (trough), or compare match on a channel other than MTU3 and MTU4.
When start requests using MTU3.TGRA compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT count.
A/D converter start requests can be specified by setting the TTGE bit to 1 in the timer interrupt enable register (TIER).
To issue an A/D converter start request at an MTU4.TCNT underflow (trough), set the TTGE2 bit in MTU4.TIER to 1.

(3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA3 (at the crest) and TCIV4 (at the trough) in MTU3 and MTU4 can be skipped up to seven times by making settings in the timer interrupt skipping set register (TITCR).
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 timer buffer transfer register (TBTER). 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 timer A/D converter request control register (TADCR).

For the linkage with the A/D converter start request delaying function, refer to section 22.3.9, A/D Converter Start Request Delaying Function.
The timer interrupt skipping setting register (TITCR) should be set while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of MTU3.TIER and MTU4.TIER under the conditions in which compare match never occur and TGIA3 and TGIA4 interrupt requests by compare match are never generated. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 22.67 shows an example of the interrupt skipping operation setting procedure. Figure 22.68 shows the periods during which interrupt skipping count can be changed.

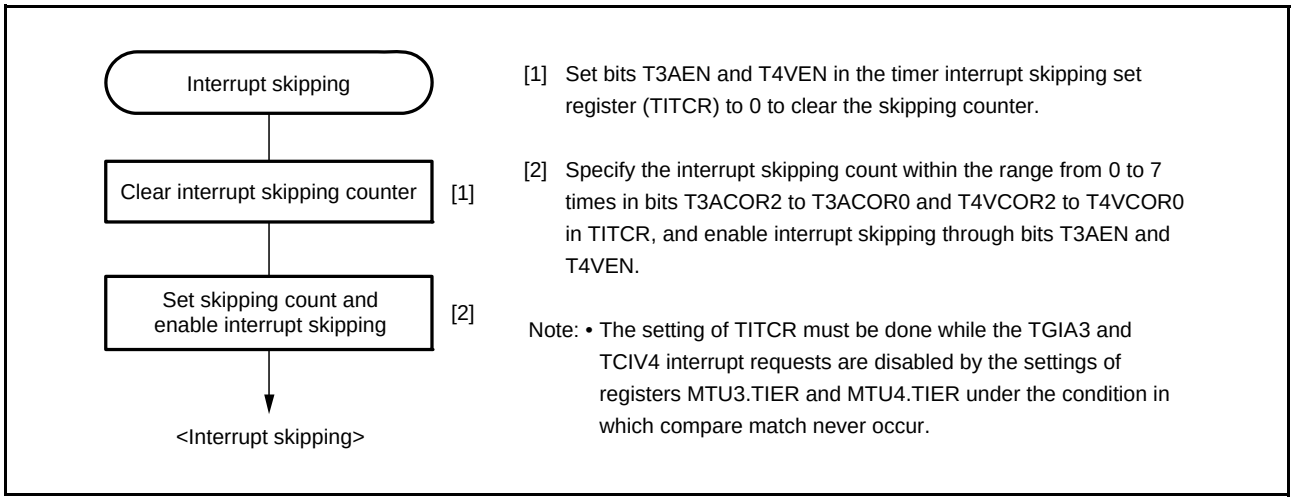


Figure 22.67 Example of Interrupt Skipping Operation Setting Procedure

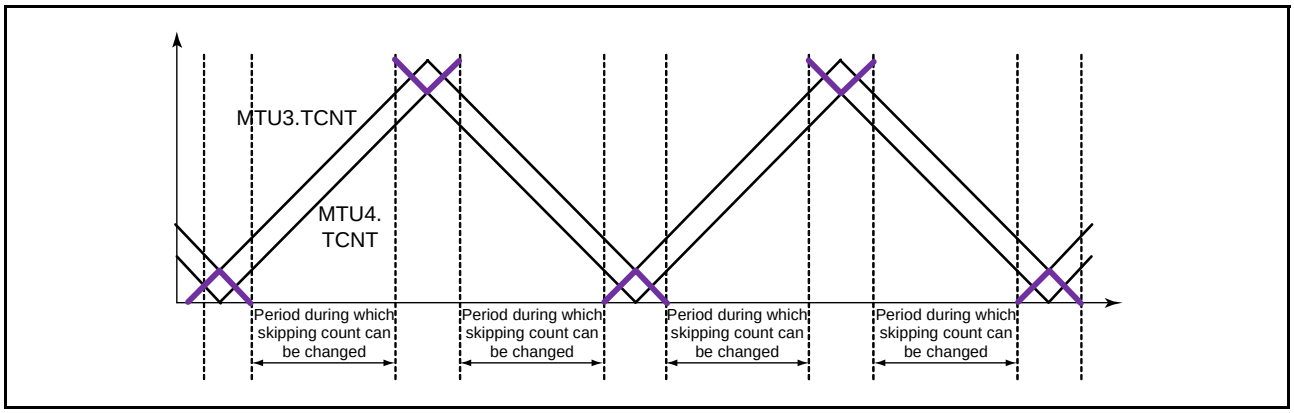


Figure 22.68 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 22.69 shows an example of MTU3.TGIA interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR bits and the T3AEN bit is set to 1 in the timer interrupt skipping set register (TITCR).

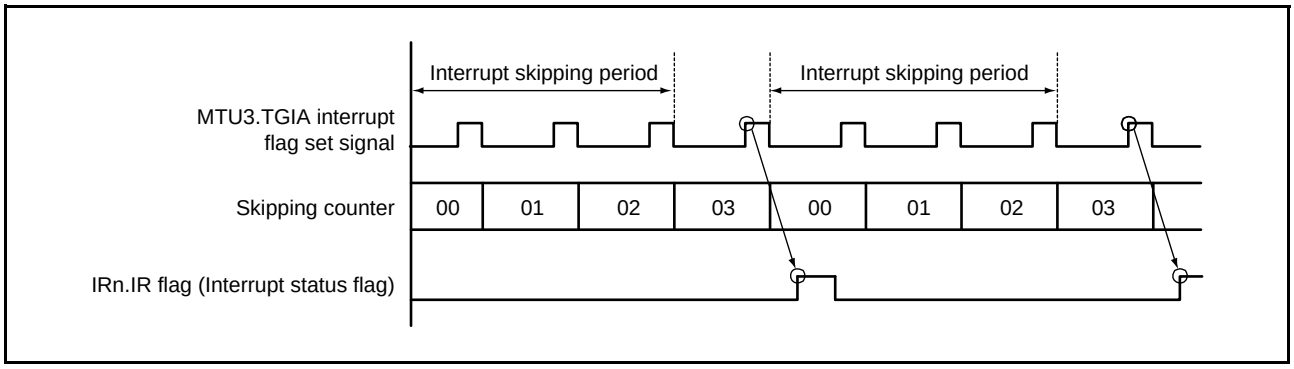


Figure 22.69 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

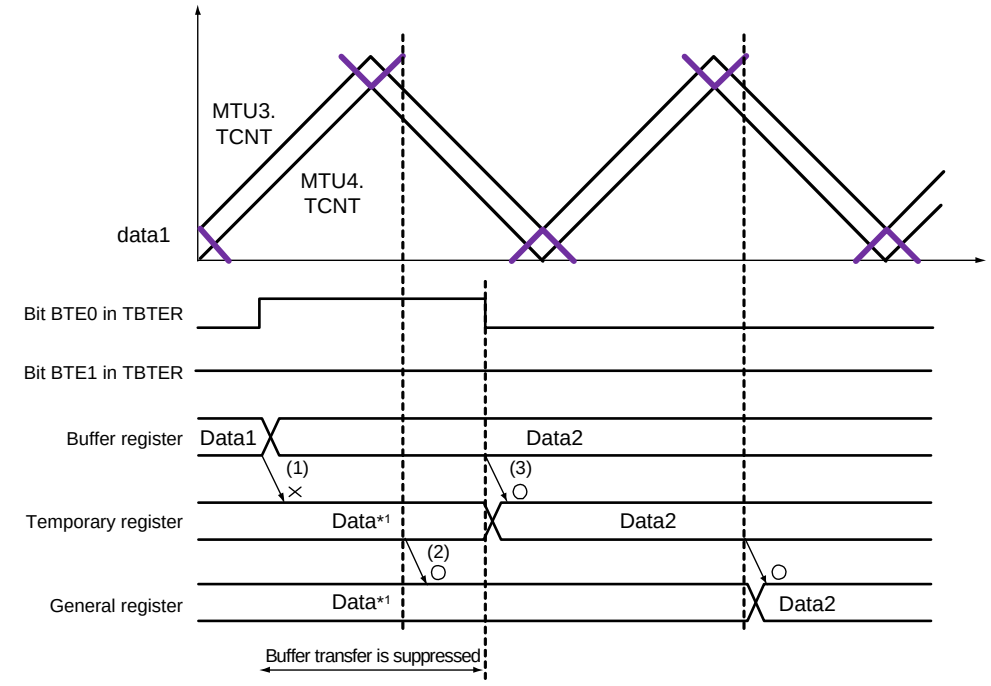
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

Figure 22.70 shows an example of operation when buffer transfer is disabled (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 22.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BET0 = 0). While this setting is valid, data is not transferred from the buffer register to the temporary register outside the buffer transfer-enabled period.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 22.72 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

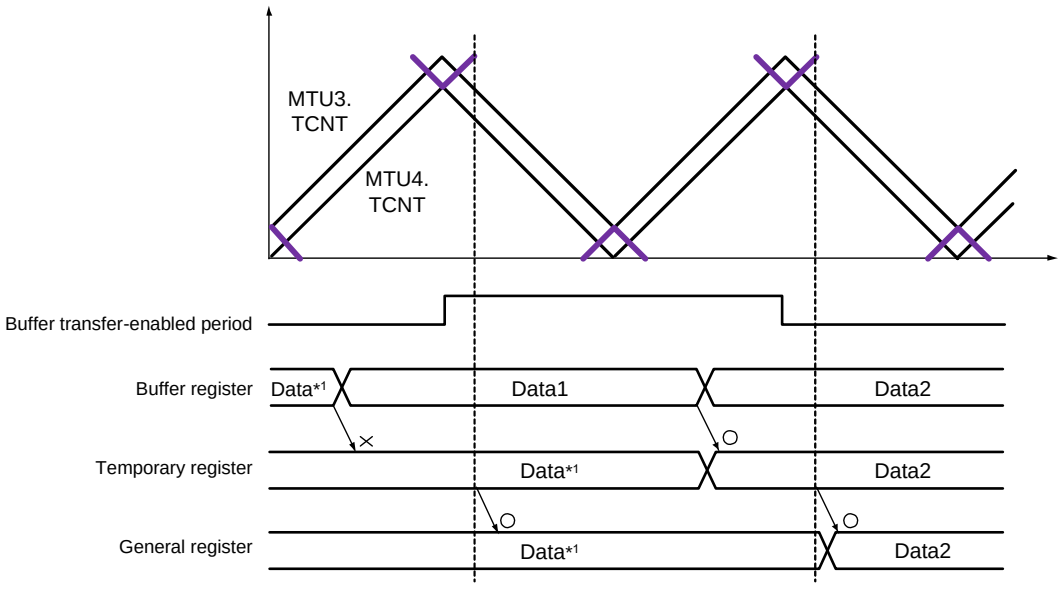
Note: • This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) 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 BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).
(2) Data is transferred from the temporary register to the general 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 22.70 Example of Operation when Buffer Transfer is Disabled (BTE1 = 0 and BTE0 = 1)



Note 1. Buffer transfer at the crest is selected.
The skipping count is set to three.
T3AEN is set to 1.

Figure 22.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)

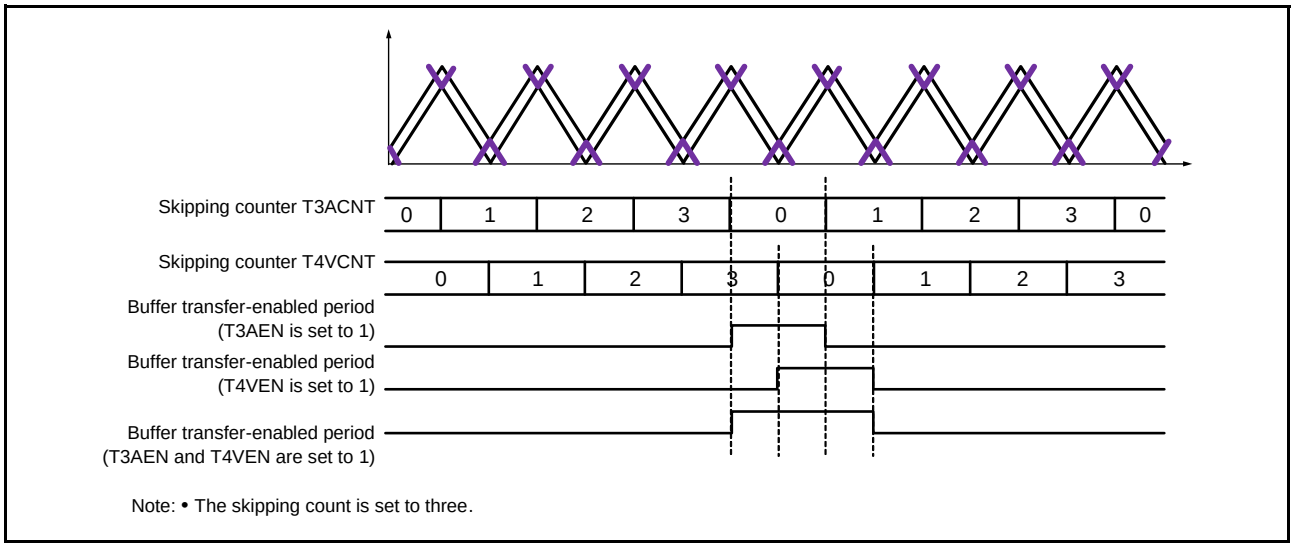


Figure 22.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Functions

The MTU2A provides the following protection functions for complementary PWM mode output.

(a) Register and Counter Miswrite Prevention Function

With the exception of the buffer registers, which can be modified at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the RWE bit in the timer read/write enable register (TRWER). The applicable registers are some of the registers in MTU3 and MTU4 shown below:

22 registers in total

MTU3.TCR and MTU4.TCR, MTU3.TMDR and MTU4.TMDR, MTU3.TIORH and MTU4.TIORH, MTU3.TIORL and MTU4.TIORL, MTU3.TIER and MTU4.TIER, MTU3.TCNT and MTU4.TCNT, MTU3.TGRA and MTU4.TGRA, MTU3.TGRB and MTU4.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, and TDDR

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 6-phase PWM output pins can be set to the high-impedance state automatically by inputting specified external signals.
See section 23, Port Output Enable 2 (POE2a), for details.

(c) Halting of PWM Output when Oscillator is Stopped

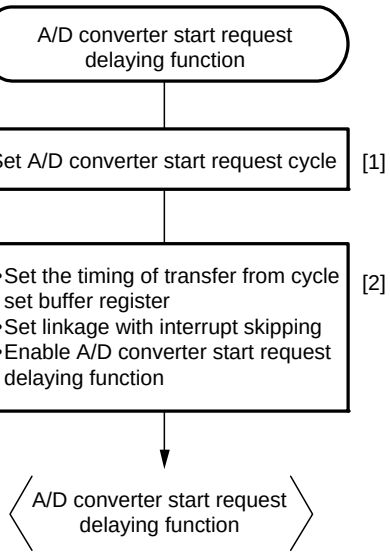
Upon detecting that the clock input to RX63N/RX631 has stopped, the 6-phase PWM output pins are automatically set to the high-impedance state. Note that the pin states are not guaranteed when the clock is restarted.
See section 9.5, Oscillation Stop Detection Function.

22.3.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in MTU4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB).
The A/D converter start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).
A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in TADCR.

(1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 22.73 shows an example of procedure for specifying the A/D converter start request delaying function.



[1] Set the cycle in the timer A/D converter start request cycle buffer register (MTU4.TADCOBRA/B) and timer A/D converter start request cycle register (MTU4.TADCORA/B). (The same initial value must be specified in the cycle buffer register and cycle register.)

[2] Use bits BF1 and BF2 in the timer A/D converter start request control register (TADCR) to specify the timing of buffer transfer from the timer A/D converter start request cycle buffer register to A/D converter start request cycle register.
- Specify whether to link with interrupt skipping through bits ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits UT4AE, DT4AE, UT4BE, and DT4BE to enable A/D converter start requests (TRG4AN or TRG4BN).

Note: • Perform TADCR setting while MTU4.TCNT is stopped.
Note: • Do not set BF1 to 1 when complementary PWM mode is not selected.
Note: • Do not set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, or DT4BE to 1 when complementary PWM mode is not selected.

Figure 22.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

(2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 22.74 shows a basic example of A/D converter start request signal (TRG4AN) operation when the trough of MTU4.TCNT is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT down-counting.

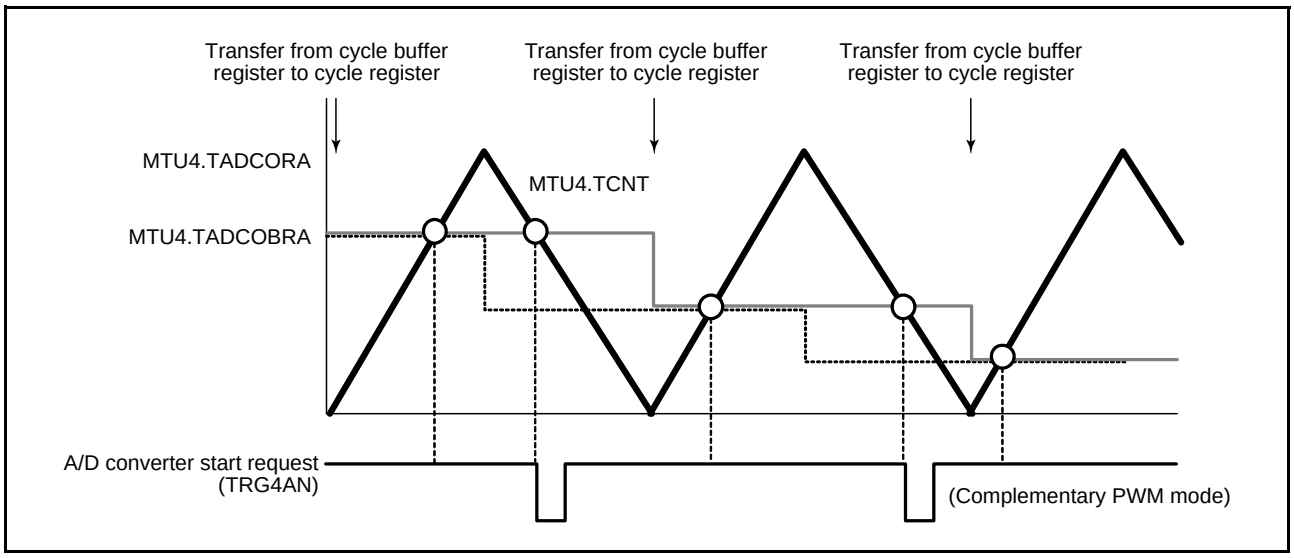


Figure 22.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

(3) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF1 and BF0 bits in the timer A/D converter start request control register (MTU4.TADCR).

(4) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR). Figure 22.75 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.
Figure 22.76 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and A/D converter start requests are linked with interrupt skipping.

Note: • This function should be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
Note that TRG4ABN (TRG4AN or TRG4BN) is output as the A/D converter start request signal in this case.

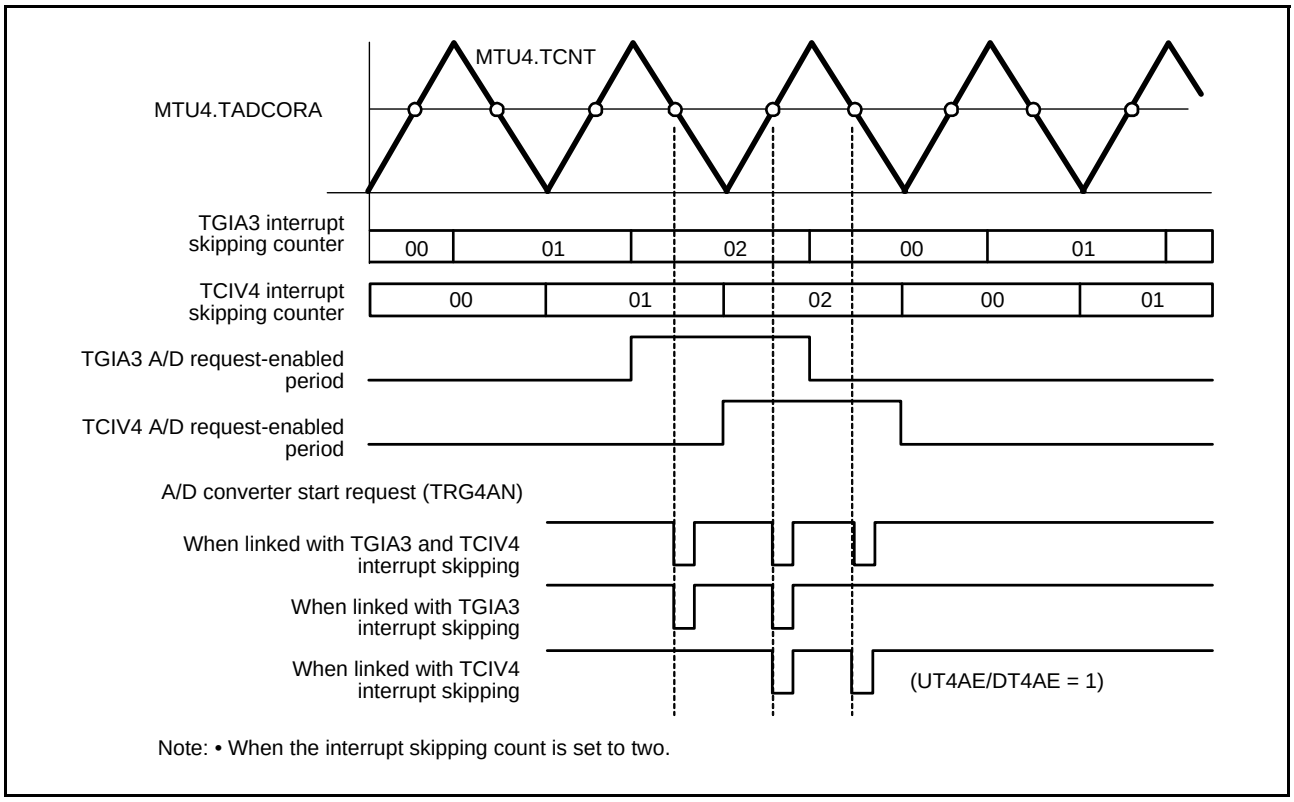


Figure 22.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up and down by TCNT is enabled)

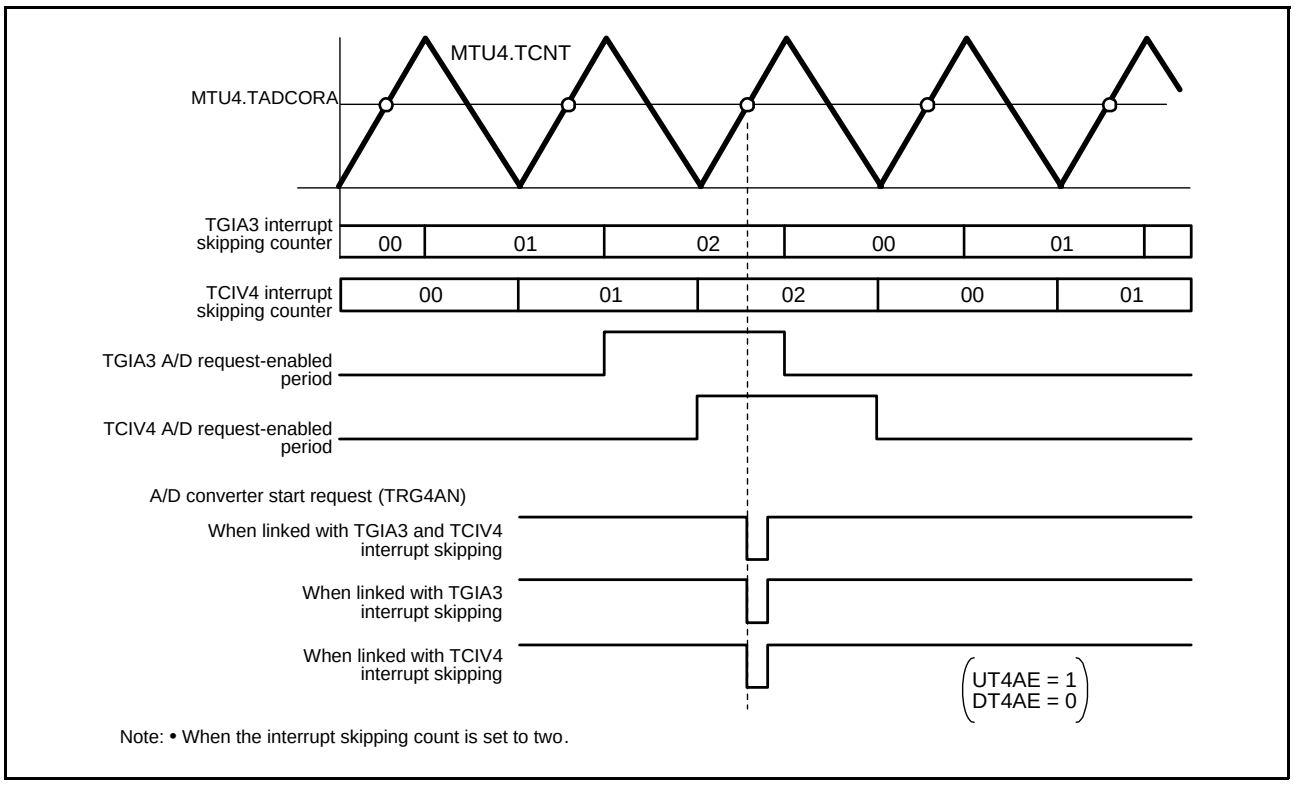


Figure 22.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up by TCNT is enabled)

22.3.10 External Pulse Width Measurement

Up to three external pulse widths can be measured in MTU5.

(1) Example of External Pulse Width Measurement Setting Procedure

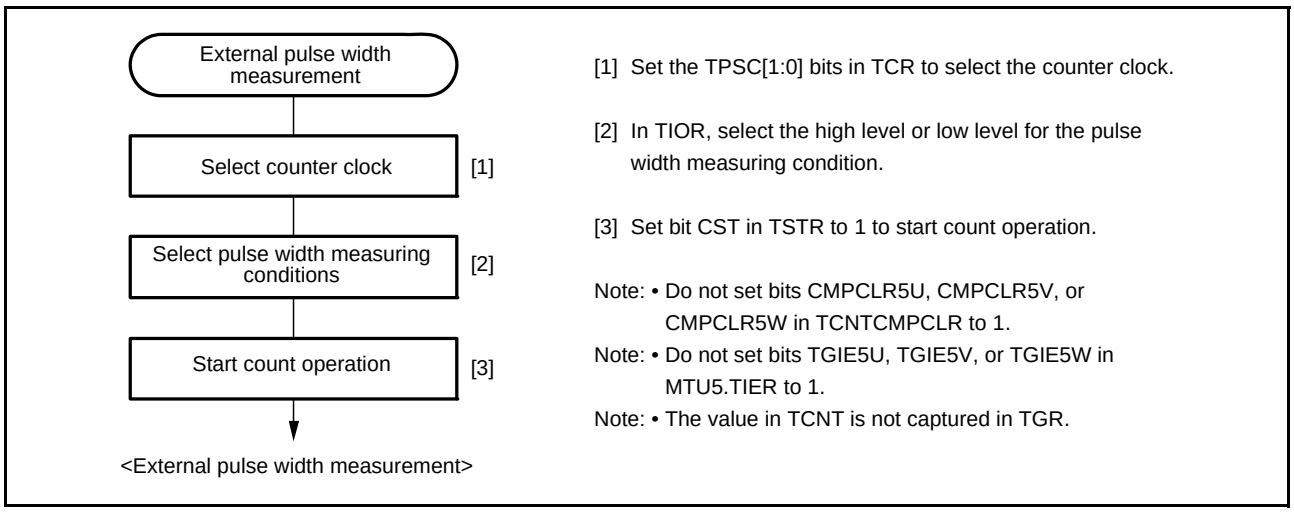


Figure 22.77 Example of External Pulse Width Measurement Setting Procedure

(2) Example of External Pulse Width Measurement

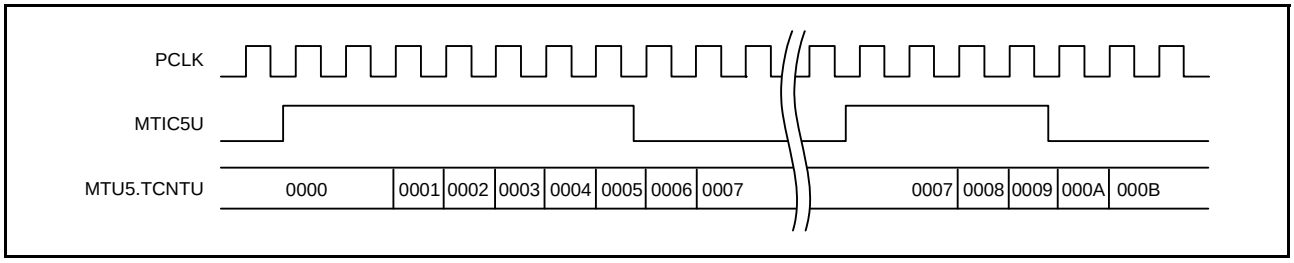


Figure 22.78 Example of External Pulse Width Measurement (Measuring High Pulse Width)

22.3.11 Dead Time Compensation

By measuring the delay of the output waveform and reflecting it to duty, the external pulse width measurement function can be used as the dead time compensation function to PWM output waveform while the complementary PWM is in operation.

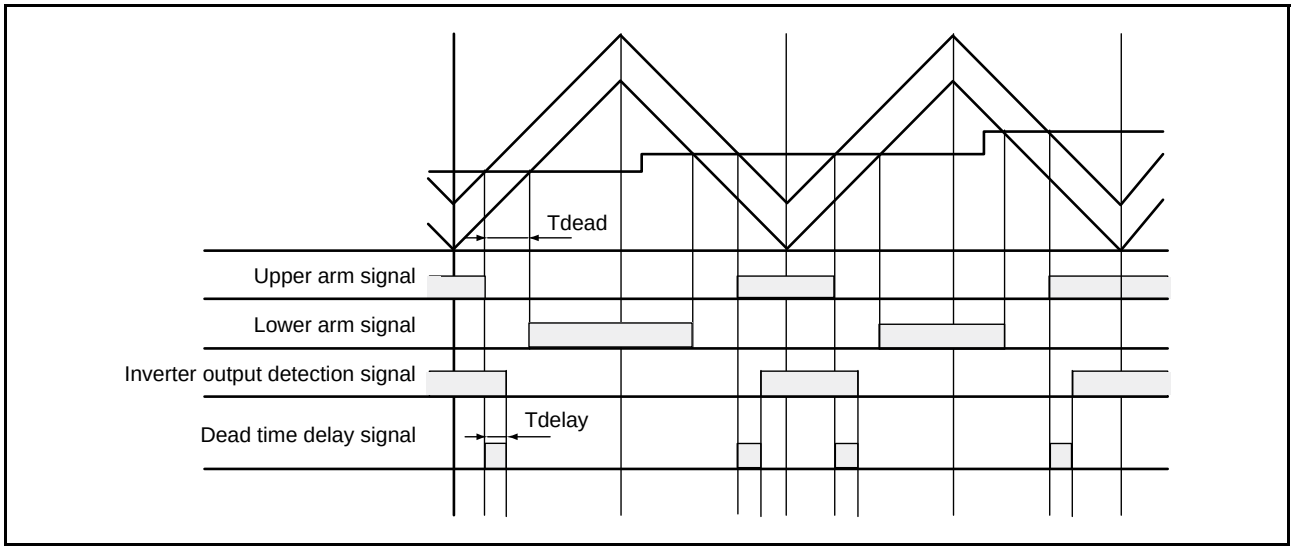
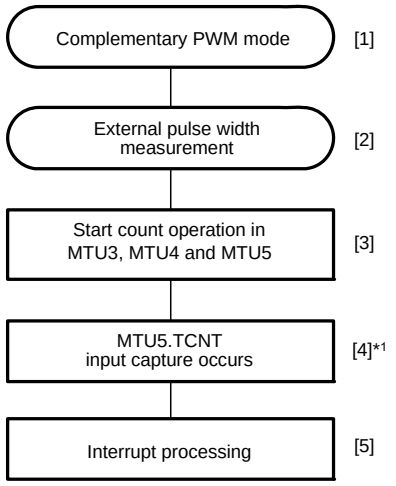


Figure 22.79 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 22.80 shows an example of dead time compensation setting procedure by using three counters in MTU5.



[1] Place MTU3 and MTU4 in complementary PWM mode. For details, refer to section, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to section, External Pulse Width Measurement.

[3] Set bits CST3 and CST4 in MTU.TSTR 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.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA3) or trough (TCIV4) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU5.TGRU and MTU3.TGRB, and write the corrected value to MTU3.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 MTU4.TGRC and MTU4.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 1. Be sure to complete PFS setting in advance.
As an interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 22.80 Example of Dead Time Compensation Setting Procedure

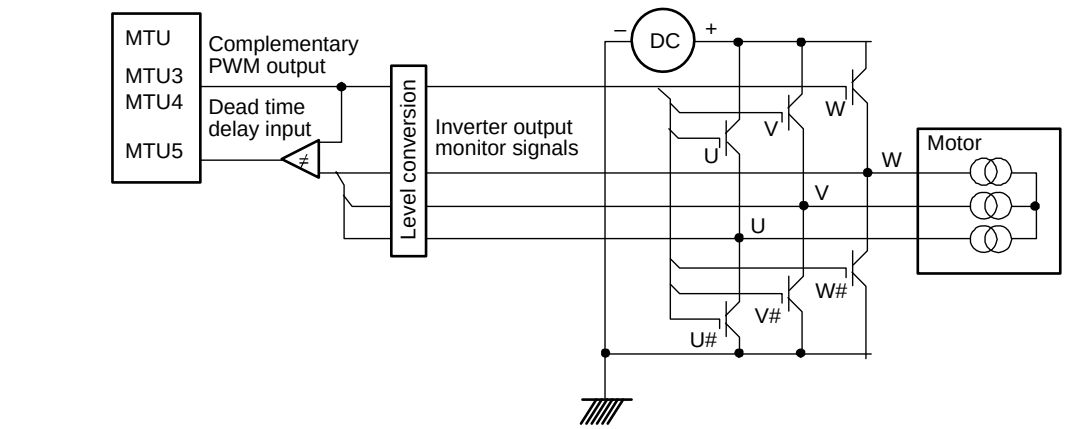


Figure 22.81 Example of Motor Control Circuit Configuration

22.3.12 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

The TCNT value is captured in TGR at either the crest or trough or at both the crest and trough during complementary PWM operation. The timing for capturing in TGR can be selected by TIOR.

Figure 22.82 is an operating example in which TCNT is used as a free-running counter without being cleared, and the TCNT value is captured in TGR at the specified timing (either crest or trough, or both crest and trough).

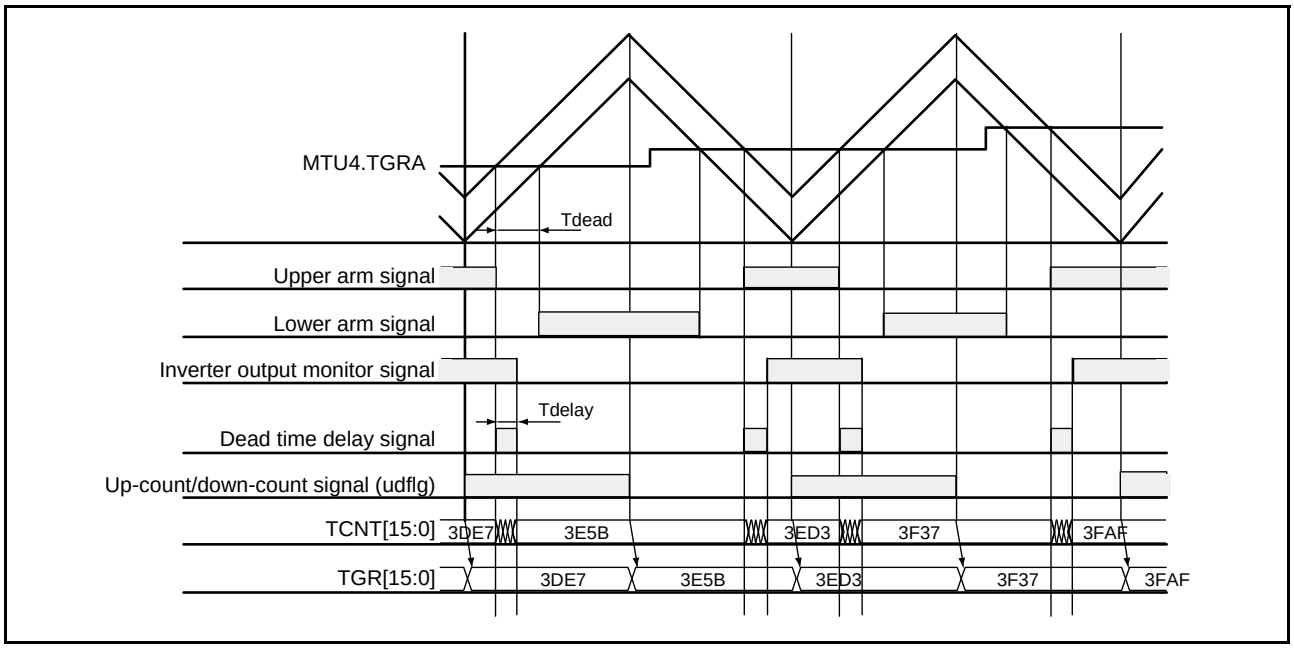


Figure 22.82 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

22.3.13 Noise Filter

Each pin for use in input capture and external pulse input to the MTU 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 functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel. Figure 22.83 shows the timing of noise filtering.

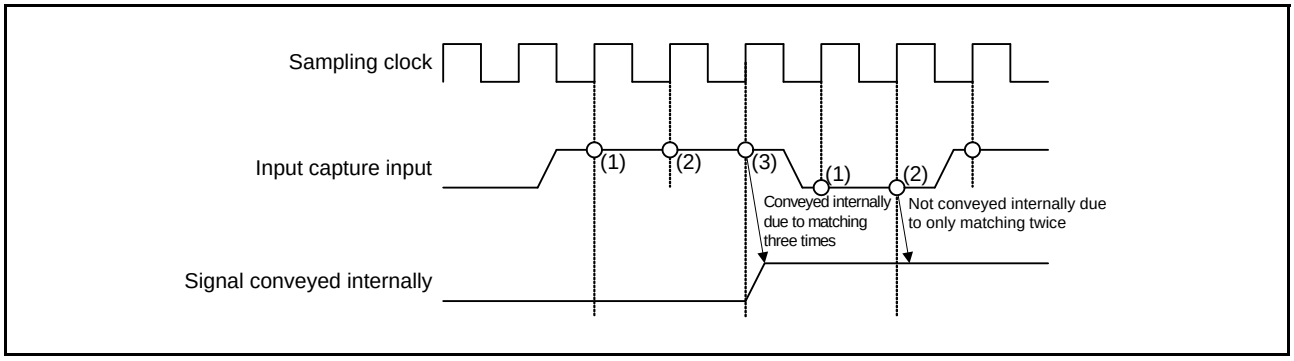


Figure 22.83 Timing of Noise Filtering

22.4 Interrupt Sources

22.4.1 Interrupt Sources and Priorities

There are three kinds of MTU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.
When an interrupt source is detected, an interrupt is requested if the corresponding enable/disable bit in TIER is set to 1.
Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, see section 15, Interrupt Controller (ICUb).
Table 22.57 lists the MTU interrupt sources.

Table 22.57 MTU Interrupt Sources (1)

Channel	Name	Interrupt Source	DMAC Activation	DTC Activation	Priority
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible	Possible	High
	TGIB0	MTU0.TGRB input capture/compare match	Not possible	Possible	
	TGIC0	MTU0.TGRC input capture/compare match	Not possible	Possible	
	TGID0	MTU0.TGRD input capture/compare match	Not possible	Possible	
	TCIV0	MTU0.TCNT overflow	Not possible	Not possible	
	TGIE0	MTU0.TGRE compare match	Not possible	Not possible	
	TGIF0	MTU0.TGRF compare match	Not possible	Not possible	
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible	Possible	
	TGIB1	MTU1.TGRB input capture/compare match	Not possible	Possible	
	TCIV1	MTU1.TCNT overflow	Not possible	Not possible	
	TCIU1	MTU1.TCNT underflow	Not possible	Not possible	
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible	Possible	
	TGIB2	MTU2.TGRB input capture/compare match	Not possible	Possible	
	TCIV2	MTU2.TCNT overflow	Not possible	Not possible	
	TCIU2	MTU2.TCNT underflow	Not possible	Not possible	
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible	Possible	
	TGIB3	MTU3.TGRB input capture/compare match	Not possible	Possible	
	TGIC3	MTU3.TGRC input capture/compare match	Not possible	Possible	
	TGID3	MTU3.TGRD input capture/compare match	Not possible	Possible	
	TCIV3	MTU3.TCNT overflow	Not possible	Not possible	
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible	Possible	
	TGIB4	MTU4.TGRB input capture/compare match	Not possible	Possible	
	TGIC4	MTU4.TGRC input capture/compare match	Not possible	Possible	
	TGID4	MTU4.TGRD input capture/compare match	Not possible	Possible	
	TCIV4	MTU4.TCNT overflow/underflow	Not possible	Possible	
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Not possible	Possible	
	TGIV5	MTU5.TGRV input capture/compare match	Not possible	Possible	
	TGIW5	MTU5.TGRW input capture/compare match	Not possible	Possible	Low

Note: • This table lists the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when a TGR input capture/compare match occurs on a channel. The MTU has 21 input capture/compare match interrupts (six for MTU0, four each for MTU3 and MTU4, two each for MTU1 and MTU2, and three for MTU5).

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when a TCNT overflow occurs on a channel. The MTU has five overflow interrupts (one for each channel).

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when a TCNT underflow occurs on a channel. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

22.4.2 DTC and 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. For details, see section 19, Data Transfer Controller (DTCa).
The MTU provides a total of 20 input capture/compare match interrupts and overflow interrupts that can be used as DTC activation sources: four each for MTU0 and MTU3, two each for MTU1 and MTU2, five for MTU4, and three for MTU5.

(2) DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 17, DMA Controller (DMACA).
The MTU provides a total of five TGRA input capture/compare match interrupts that can be used as DMAC activation sources: one each for MTU0 to MTU4.
When the DMAC is activated by the MTU, the activation source is cleared when the DMAC requests the internal bus mastership. Therefore, the request for DMAC transfer may be kept pending for a certain period even after the activation source is cleared depending on the internal bus state.

22.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following five methods in the MTU. Table 22.58 lists 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 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 is set to 1, the A/D converter can be activated at the trough of MTU4.TCNT count (MTU4.TCNT = 0000h).
A/D converter start request signal TRGAN is issued to the A/D converter under either of the following conditions.

- When a TGRA input capture/compare match occurs on a channel while the TTGE bit in TIER is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM operation while the TTGE2 bit in MTU4.TIER is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN 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 compare match between MTU0.TCNT and MTU0.TGRE activates the A/D converter.
A/D converter start request signal TRG0EN is issued when a compare match occurs between MTU0.TCNT and MTU0.TGRE. If A/D converter start signal TRG0EN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRF

A compare match between MTU0.TCNT and MTU0.TGRF activates the A/D converter.
A/D converter start request signal TRG0FN is issued when a compare match occurs between MTU0.TCNT and MTU0.TGRF. If A/D converter start signal TRG0FN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(4) A/D Converter Activation by Input Capture or Compare Match with MTU0.TGRA, TGRB, TGRC or TGRD

The A/D converter can be activated when an input capture or compare match occurs between MTU0.TCNT and MTU0.TGRA, MTU0.TGRB, MTU0.TGRC or MTU0.TGRD.
When an input capture or compare match occurs between MTU0.TCNT and MTU0.TGRA, MTU0.TGRB, MTU0.TGRC or MTU0.TGRD. A/D converter start request signal TRG0AN, TRG0BN, TRG0CN or TRG0DN is issued. If A/D converter start signal TRG0AN, TRG0BN, TRG0CN or TRG0DN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(5) 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 when the MTU4.TCNT count matches the TADCORA or TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 22.3.9, A/D Converter Start Request Delaying Function.
A/D conversion will start if A/D converter start signal TRG4ABN from the MTU is selected as the trigger in the A/D converter when TRG4AN or TRG4BN is generated.

Table 22.58 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	A/D Start Request Source	A/D Converter Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGAN
MTU1.TGRA and MTU1.TCNT		
MTU2.TGRA and MTU2.TCNT		
MTU3.TGRA and MTU3.TCNT		
MTU4.TGRA and MTU4.TCNT		
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRG0AN
MTU0.TGRB and MTU0.TCNT		TRG0BN
MTU0.TGRC and MTU0.TCNT		TRG0CN
MTU0.TGRD and MTU0.TCNT		TRG0DN
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0EN
MTU0.TGRF and MTU0.TCNT		TRG0FN
TADCORA and MTU4.TCNT or TADCORB and MTU4.TCNT		TRG4ABN

22.5 Operation Timing

22.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 22.84 and Figure 22.85 show the TCNT count timing for TGI interrupt in internal clock operation, Figure 22.86 shows the TCNT count timing in external clock operation (normal mode), and Figure 22.87 shows the TCNT count timing in external clock operation (phase counting mode).

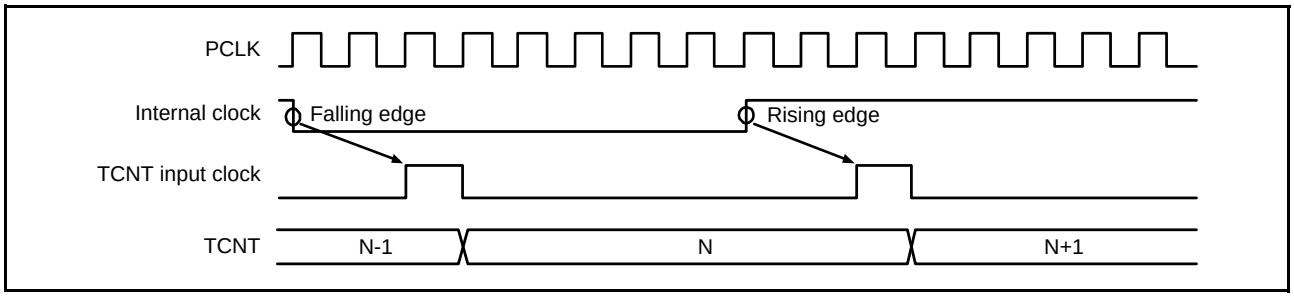


Figure 22.84 Count Timing in Internal Clock Operation (MTU0 to MTU4)

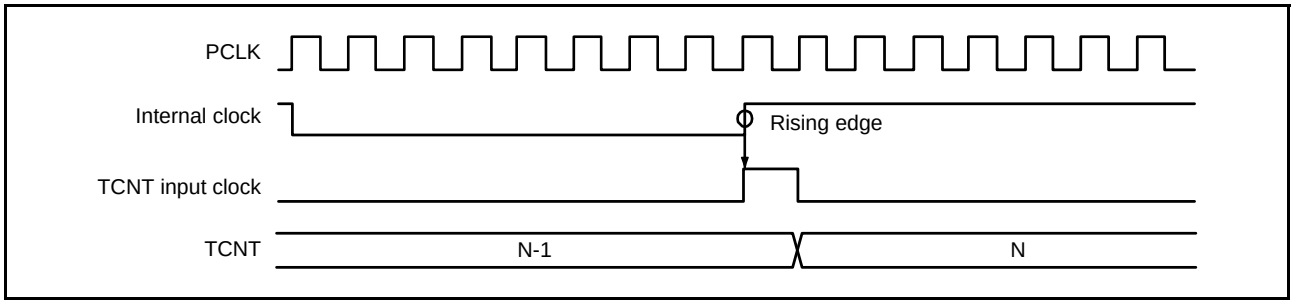


Figure 22.85 Count Timing in Internal Clock Operation (MTU5)

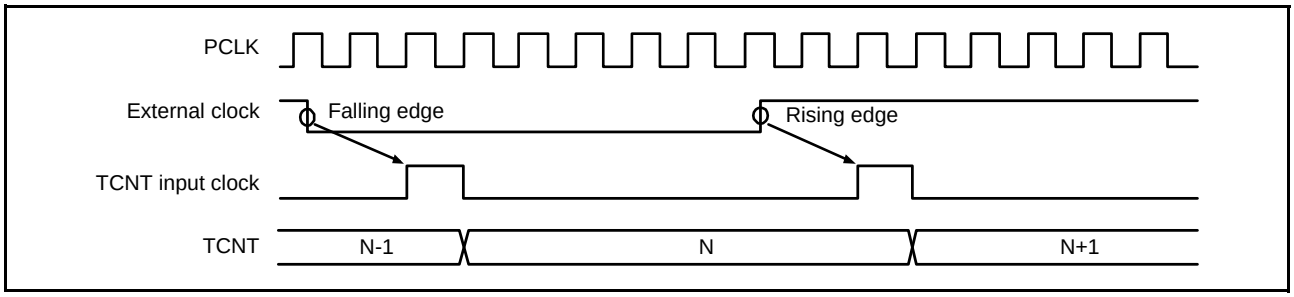


Figure 22.86 Count Timing in External Clock Operation (MTU0 to MTU 4)

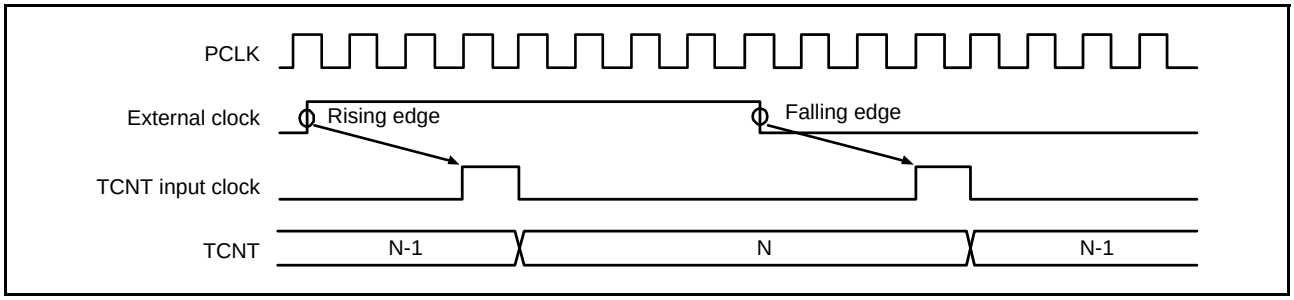


Figure 22.87 Count Timing in External Clock Operation (Phase Counting Mode)

(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 is updated by TCNT). When a compare match signal is generated, the value set in TIOR is output to the output compare output pin (MTIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 22.88 shows the output compare output timing (normal mode or PWM mode) and Figure 22.89 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

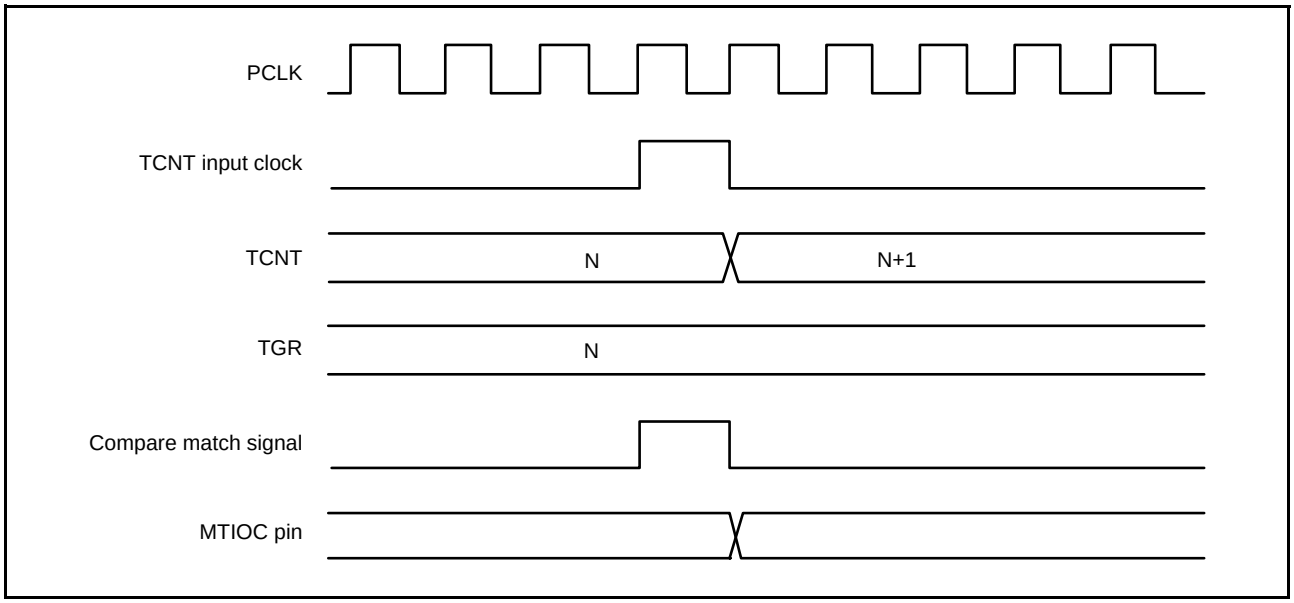


Figure 22.88 Output Compare Output Timing (Normal Mode or PWM Mode)

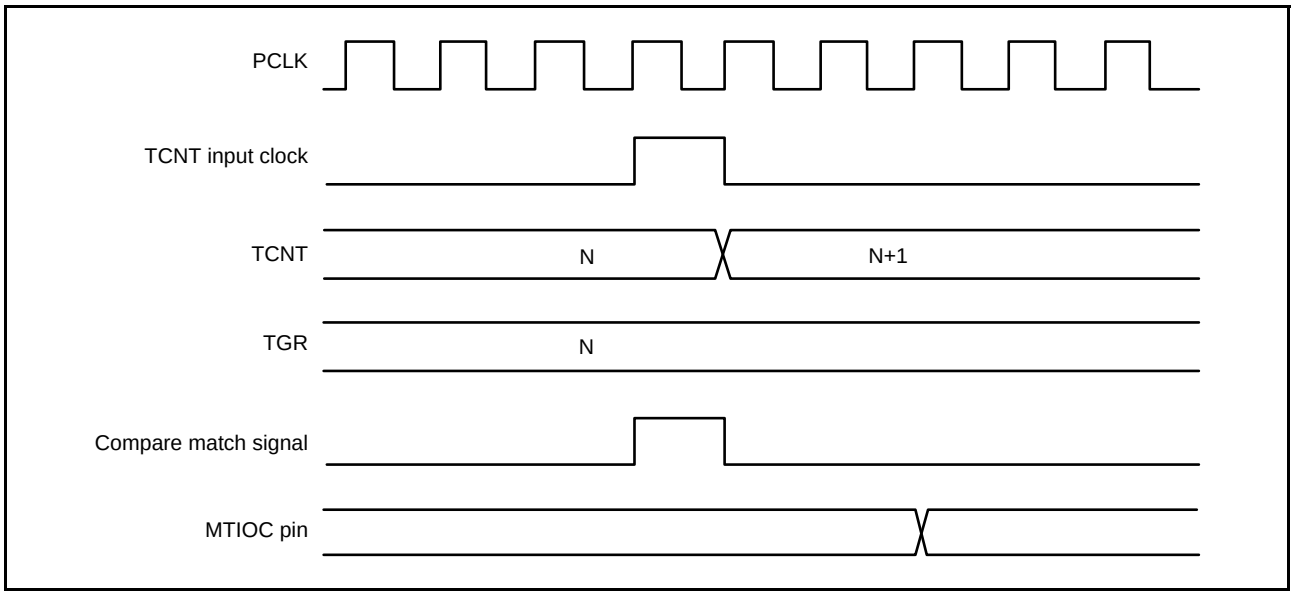


Figure 22.89 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode)

(3) Input Capture Signal Timing

Figure 22.90 shows the input capture signal timing.

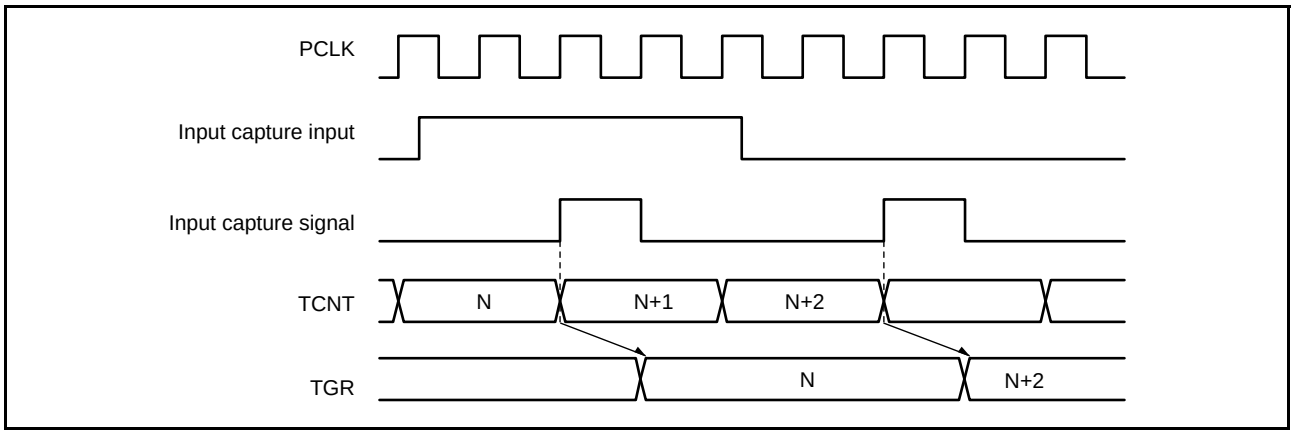


Figure 22.90 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 22.91 and Figure 22.92 show the timing when counter clearing on compare match is specified, and Figure 22.93 shows the timing when counter clearing on input capture is specified.

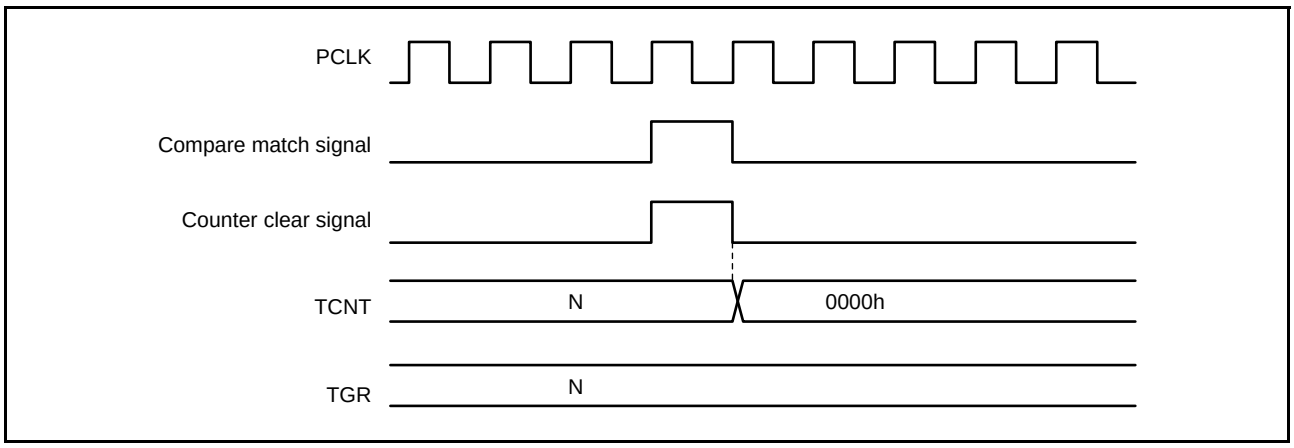


Figure 22.91 Counter Clear Timing (Compare Match) (MTU0 to MTU4)

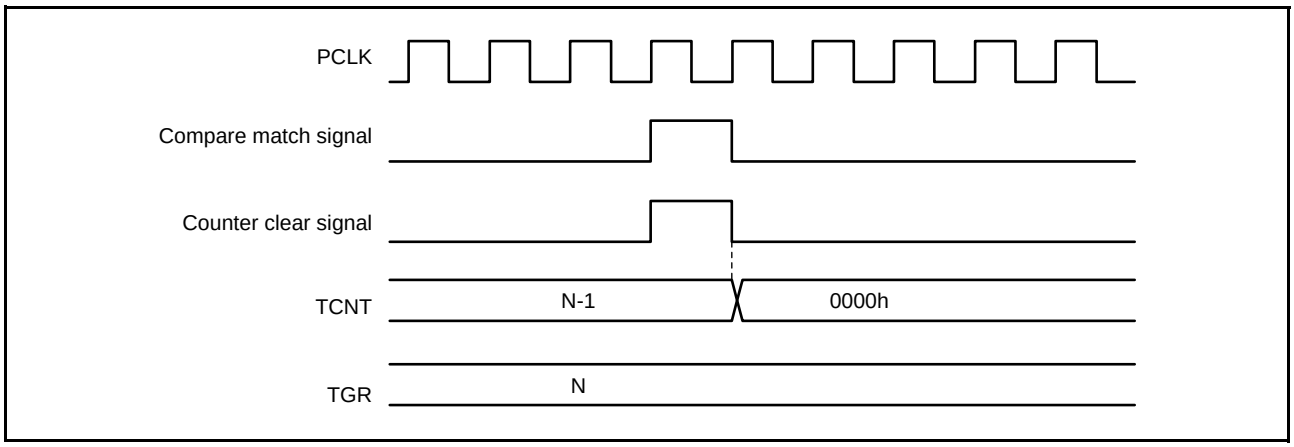


Figure 22.92 Counter Clear Timing (Compare Match) (MTU5)

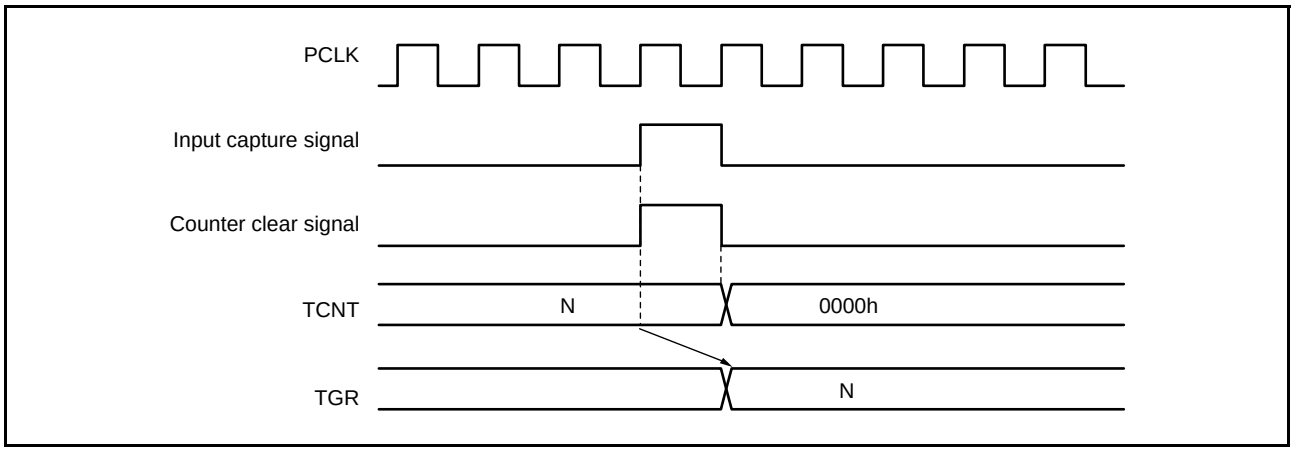


Figure 22.93 Counter Clear Timing (Input Capture) (MTU0 to MTU5)

(5) Buffer Operation Timing

Figure 22.94 to Figure 22.96 show the timing in buffer operation.

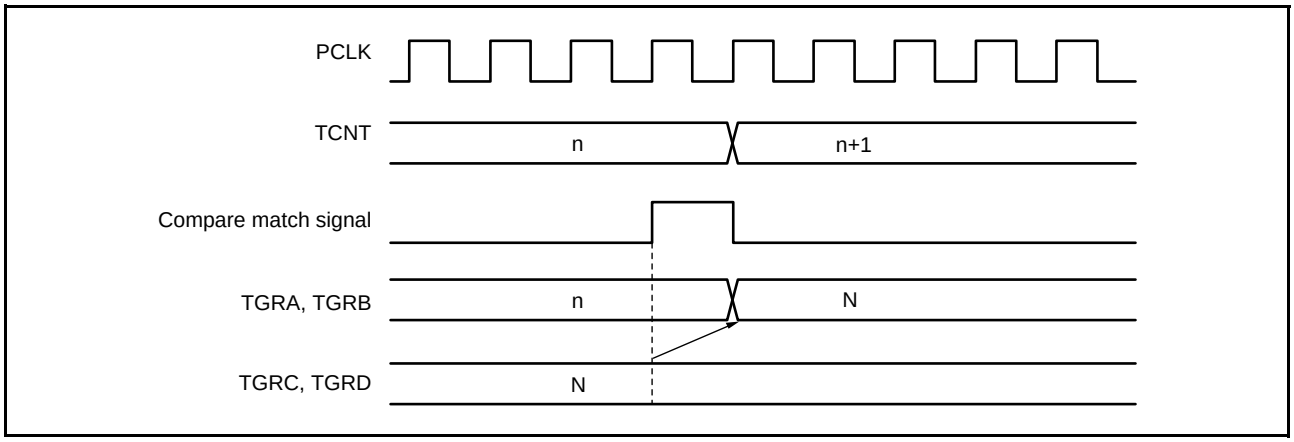


Figure 22.94 Buffer Operation Timing (Compare Match)

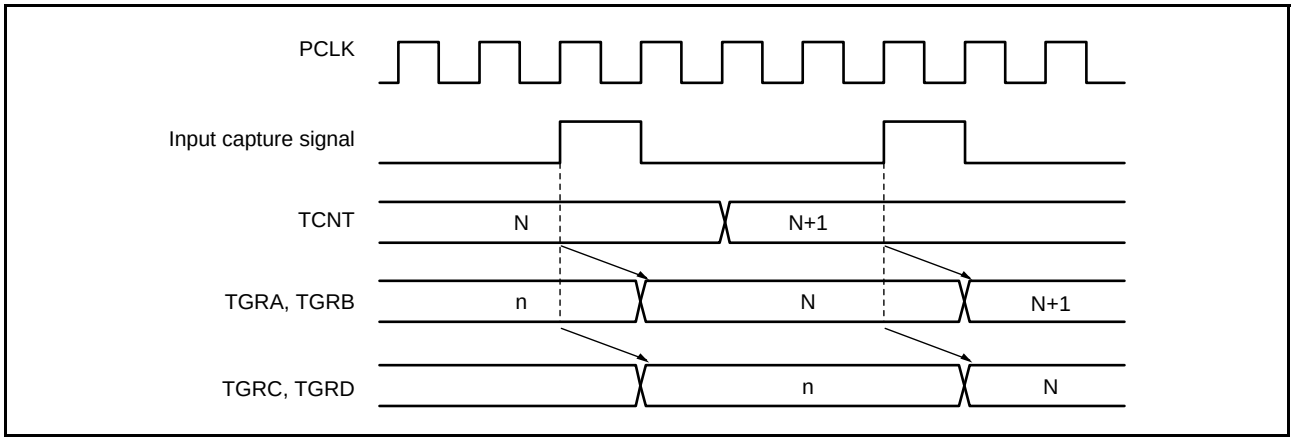


Figure 22.95 Buffer Operation Timing (Input Capture)

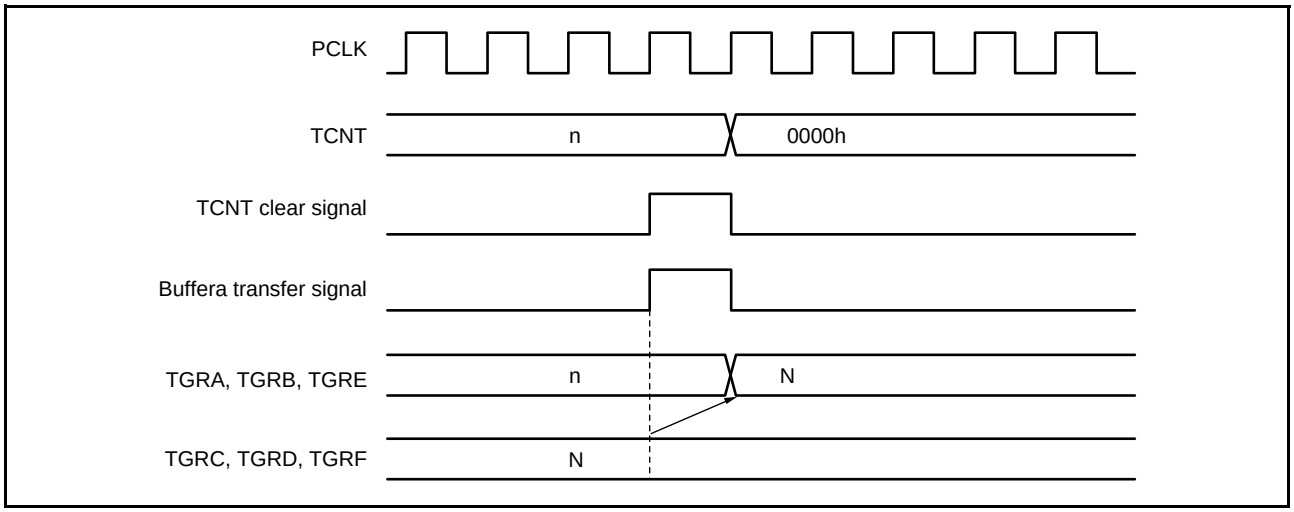


Figure 22.96 Buffer Operation Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 22.97 to Figure 22.99 show the buffer transfer timing in complementary PWM mode.

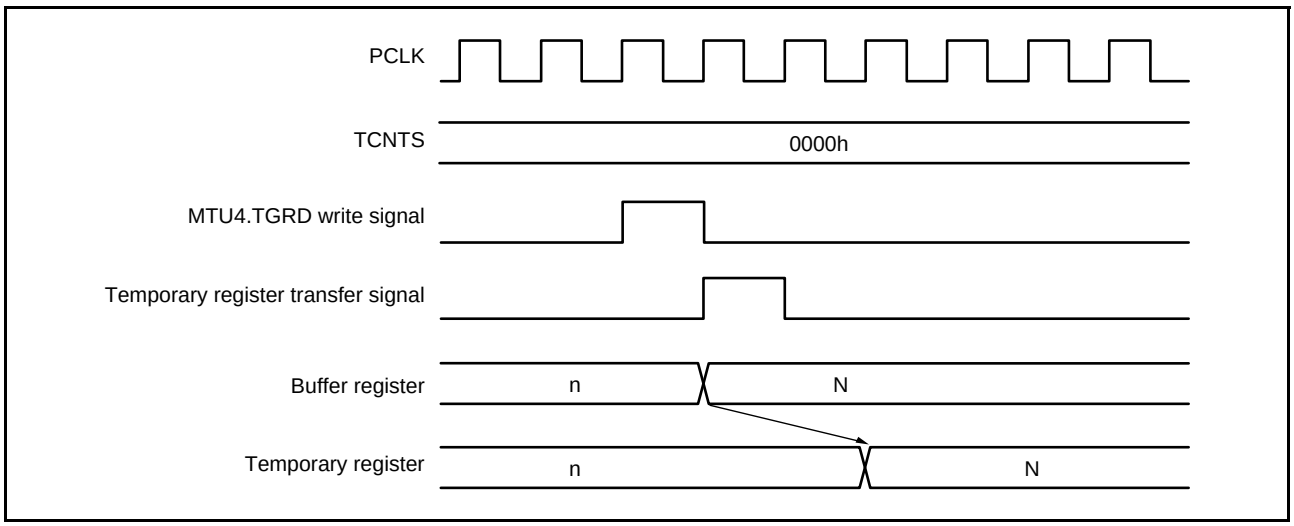


Figure 22.97 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

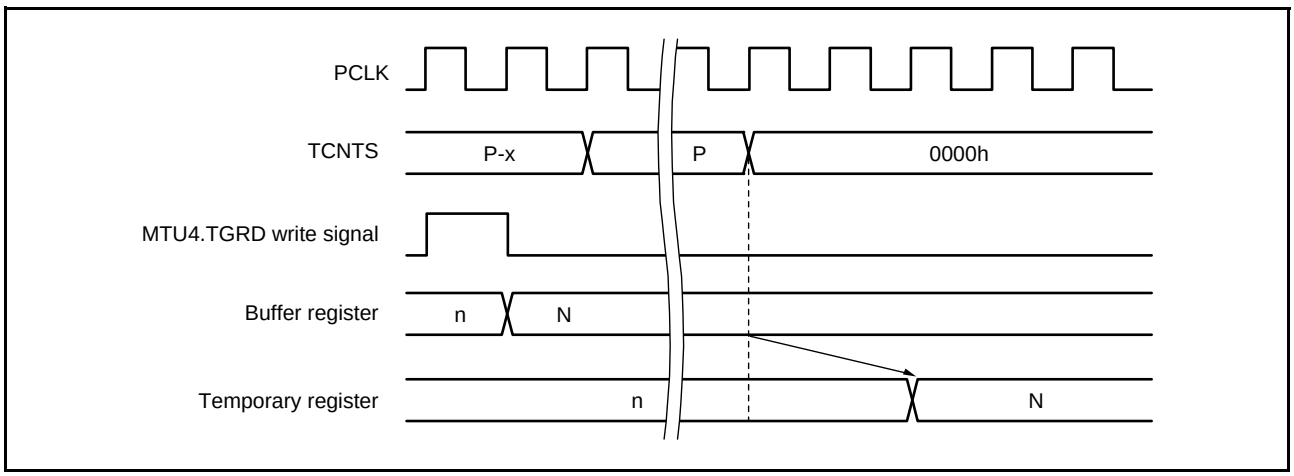


Figure 22.98 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

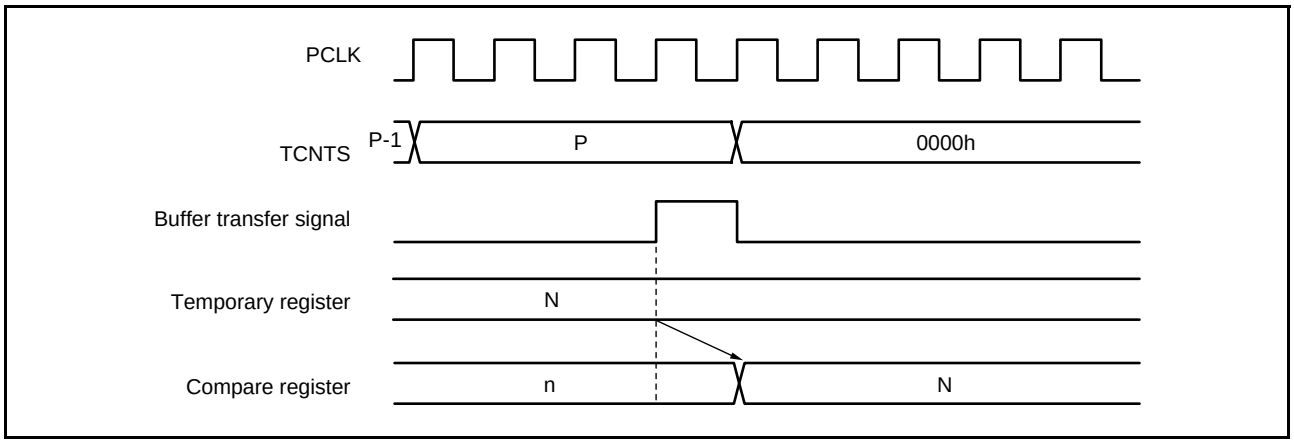


Figure 22.99 Transfer Timing from Temporary Register to Compare Register

22.5.2 Interrupt Signal Timing

(1) Timing for TGI Interrupt by Compare Match

Figure 22.100 and Figure 22.101 show the TGI interrupt request signal timing on compare match.

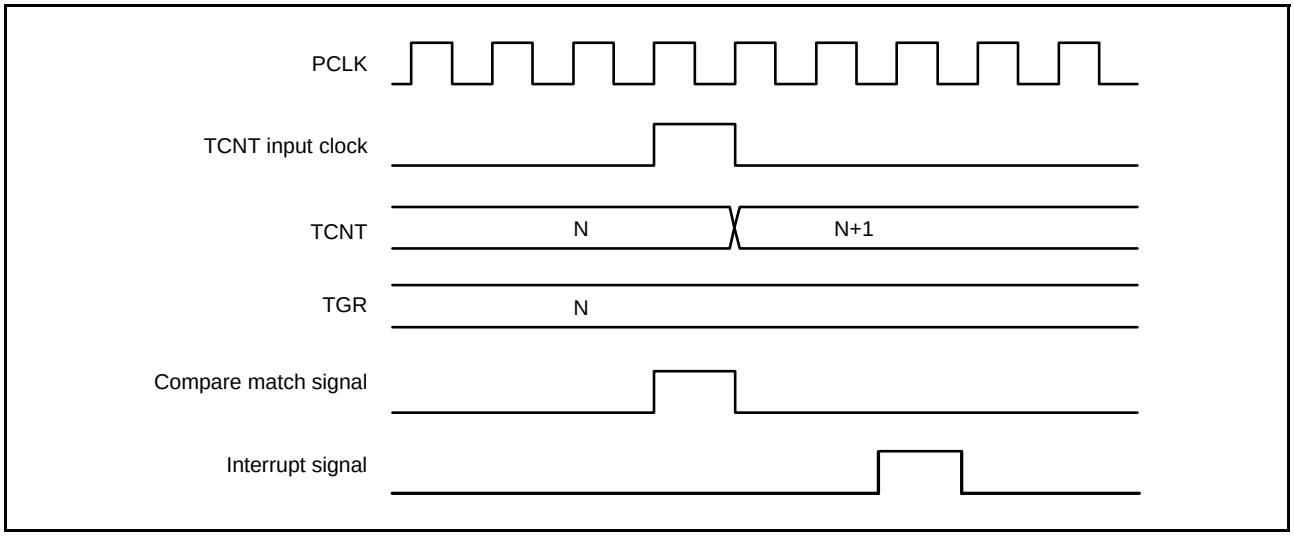


Figure 22.100 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)

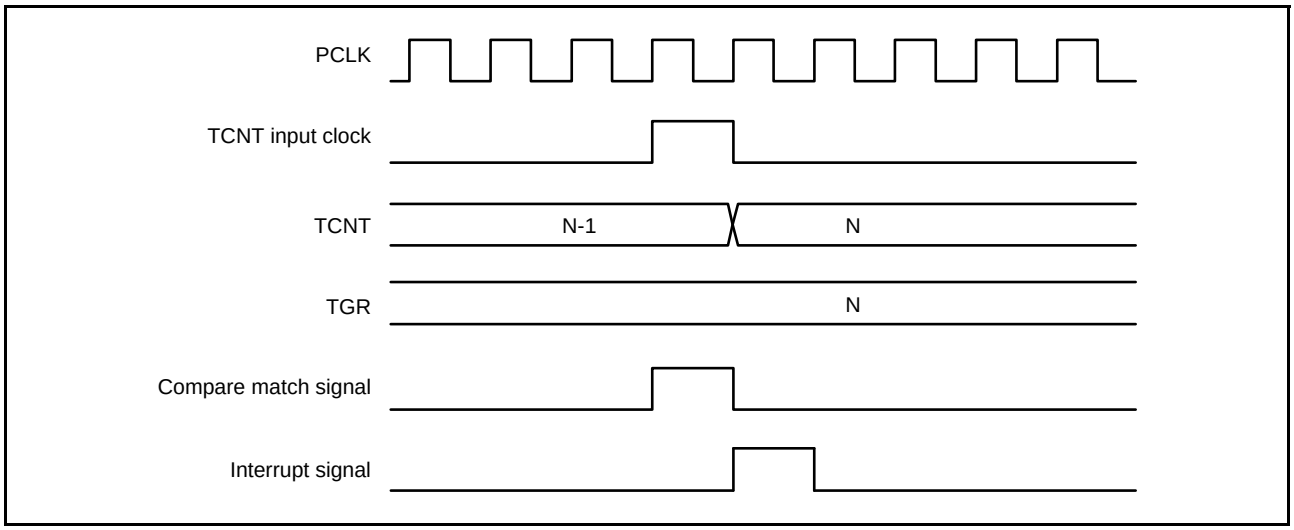


Figure 22.101 TGI Interrupt Timing (Compare Match) (MTU5)

(2) Timing for TGI Interrupt Setting by Input Capture

Figure 22.102 and Figure 22.103 show TGI interrupt request signal timing on input capture.

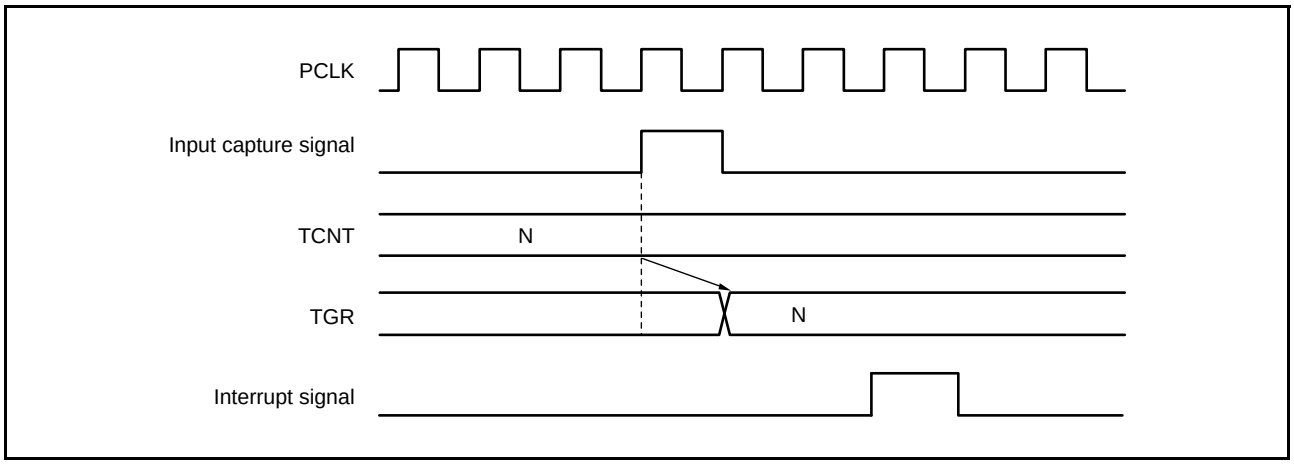


Figure 22.102 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)

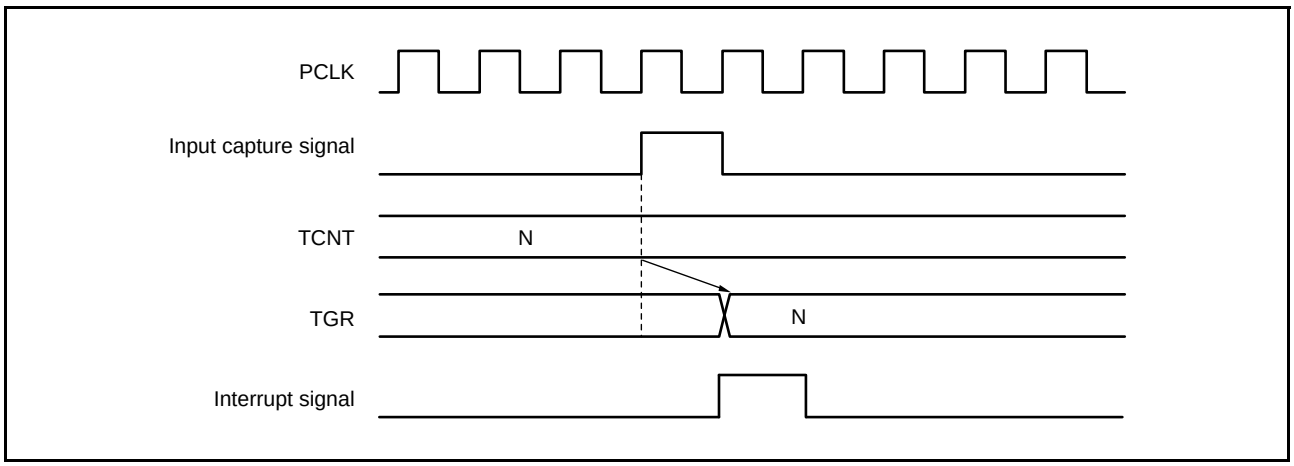


Figure 22.103 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 22.104 shows the TCIV interrupt request signal timing on overflow.
Figure 22.105 shows the TCIU interrupt request signal timing on underflow.

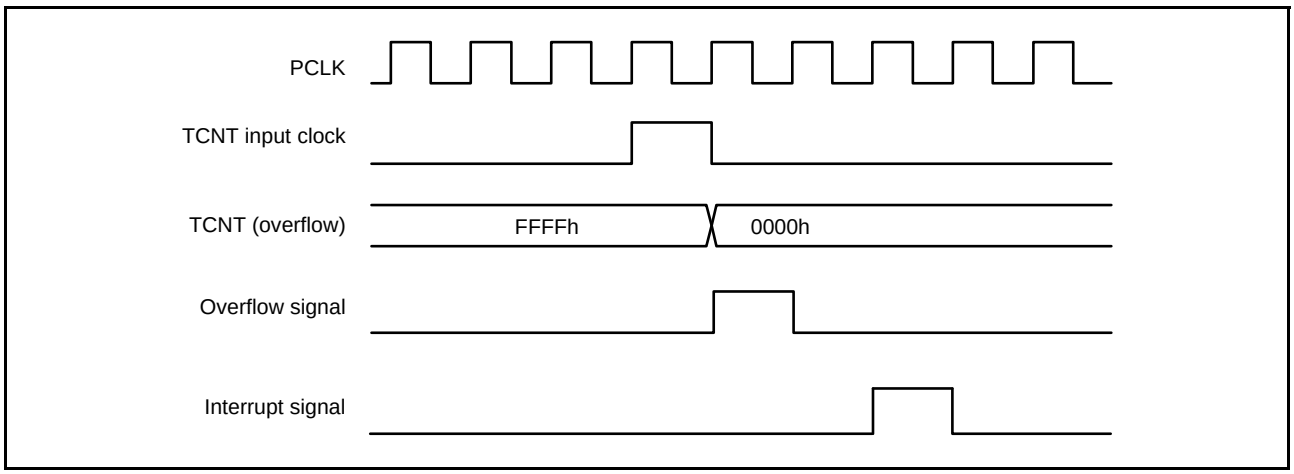


Figure 22.104 TCIV Interrupt Setting Timing

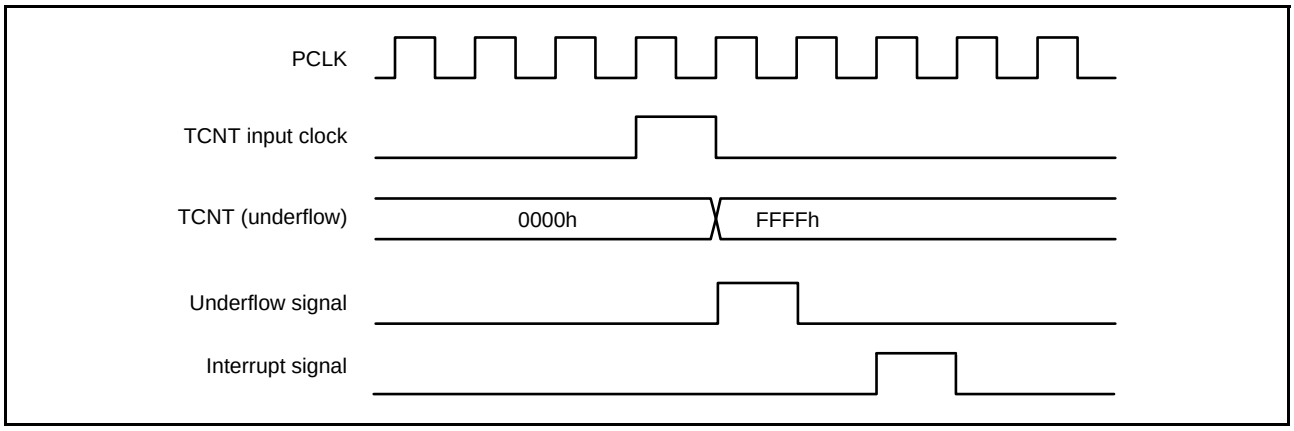


Figure 22.105 TCIU Interrupt Setting Timing

22.6 Usage Notes

22.6.1 Module Clock Stop Mode 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 clearing the module clock stop mode. For details, refer to section 11, Low Power Consumption.

22.6.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK cycles for single-edge detection, and at least 2.5 PCLK cycles for both-edge detection. The MTU2A 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 1.5 PCLK, and the pulse width must be at least 2.5 PCLK cycles. Figure 22.106 shows the input clock conditions in phase counting mode.

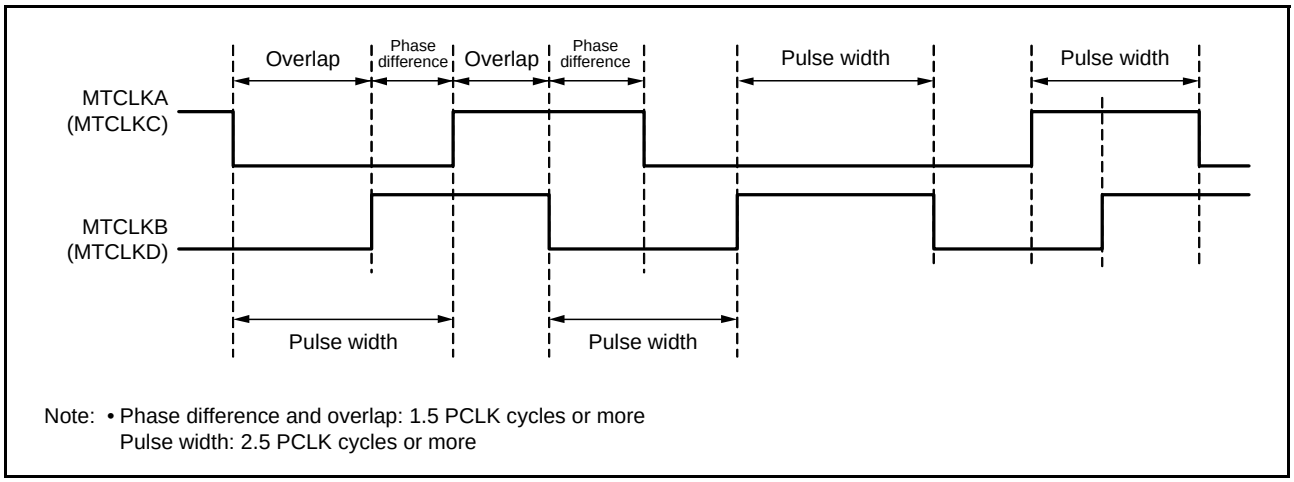


Figure 22.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

22.6.3 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

$$f = \frac{PCLK}{(N+1)}$$

- MTU5

$$f = \frac{PCLK}{N}$$

f: Counter frequency
PCLK: MTU clock operating frequency
N: TGR setting

22.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in a TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.
Figure 22.107 shows the timing in this case.

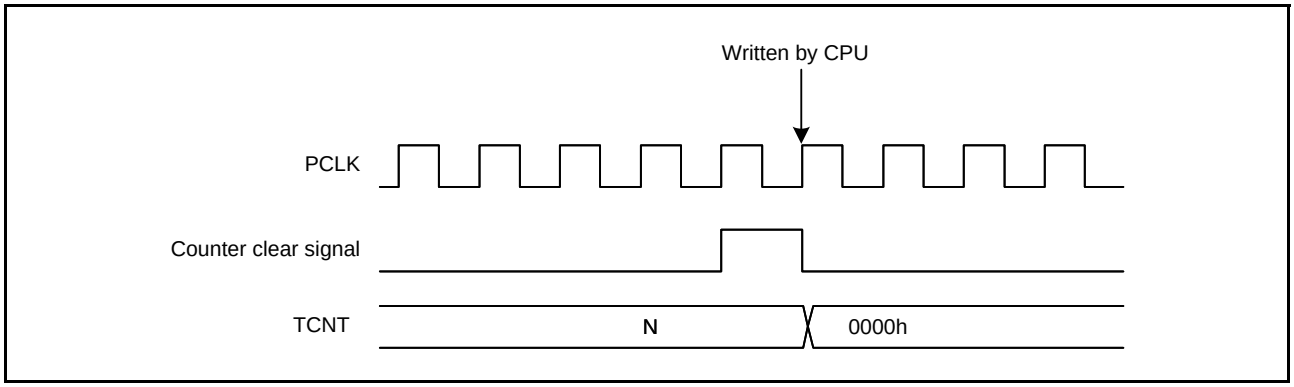


Figure 22.107 Contention between TCNT Write and Clear Operations

22.6.5 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 22.108 shows the timing in this case.

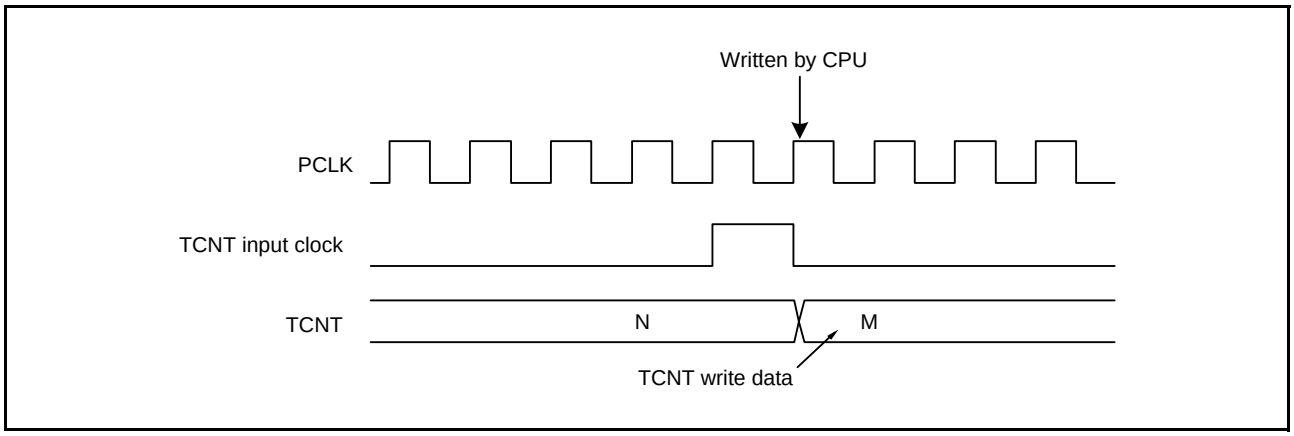


Figure 22.108 Contention between TCNT Write and Increment Operations

22.6.6 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 22.109 shows the timing in this case.

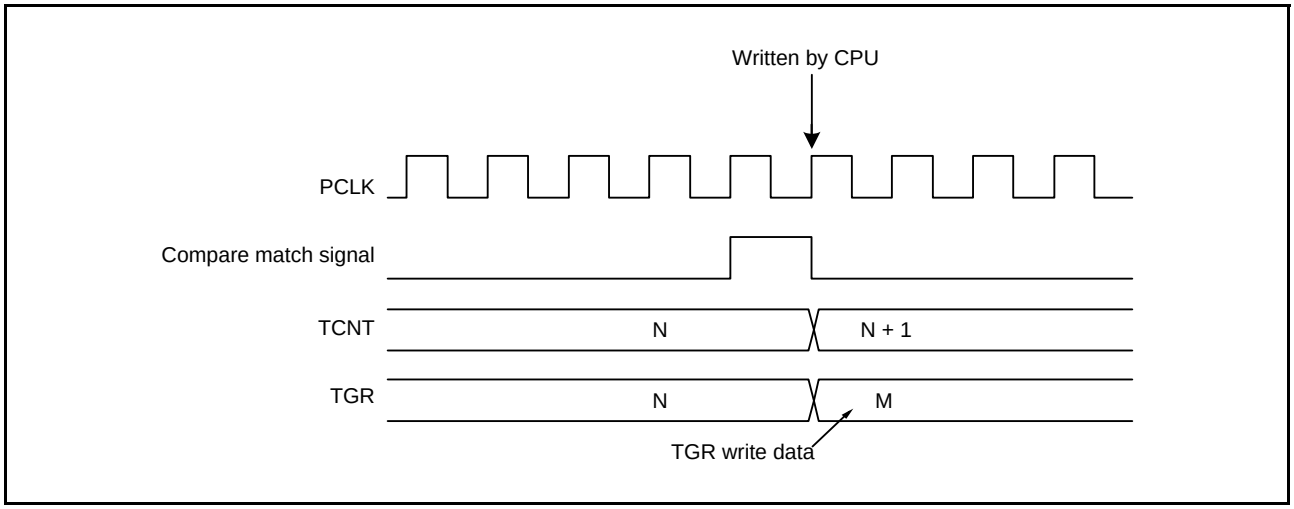


Figure 22.109 Contention between TGR Write Operation and Compare Match

22.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 22.110 shows the timing in this case.

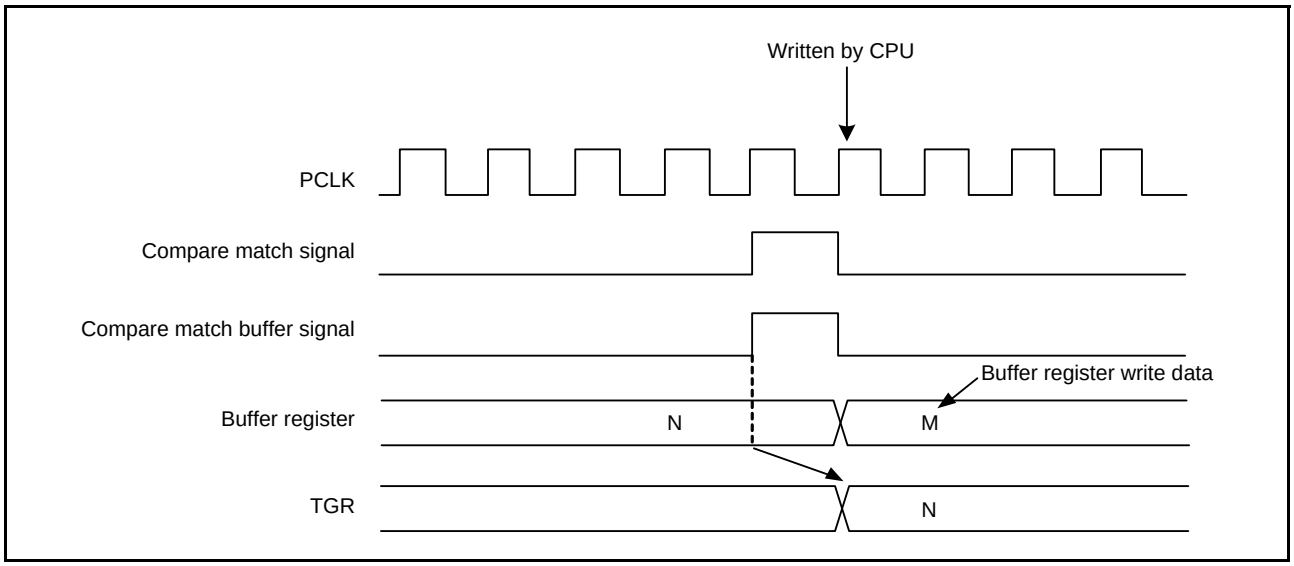


Figure 22.110 Contention between Buffer Register Write Operation and Compare Match

22.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the buffer transfer mode register (TBTM), if TCNT clearing occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation. Figure 22.111 shows the timing in this case.

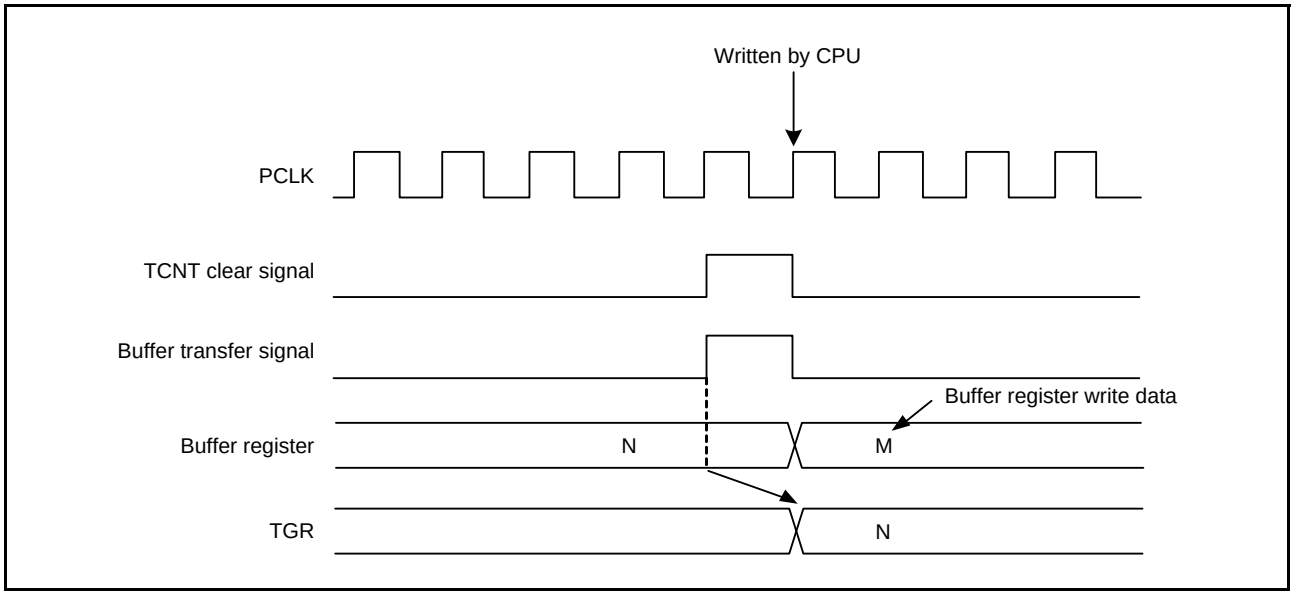


Figure 22.111 Contention between Buffer Register Write and TCNT Clear Operations

22.6.9 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 in MTU0 to MTU4, and the data after input capture transfer is read in MTU5.
Figure 22.112 and Figure 22.113 show the timing in this case.

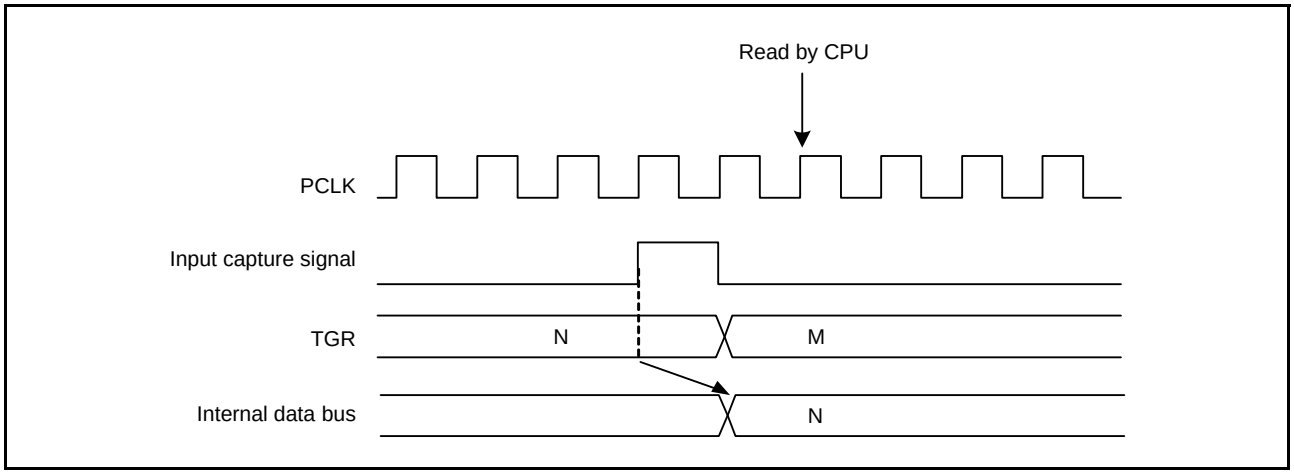


Figure 22.112 Contention between TGR Read Operation and Input Capture (MTU0 to MTU4)

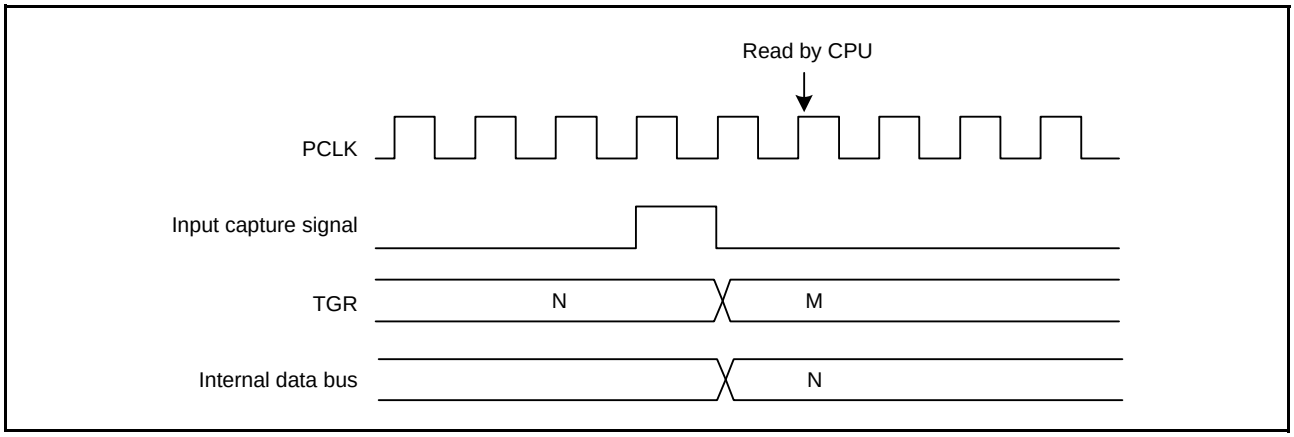


Figure 22.113 Contention between TGR Read Operation and Input Capture (MTU5)

22.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in a TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4. In MTU5, the TGR write operation is performed and the input capture signal is generated.

Figure 22.114 and Figure 22.115 show the timing in this case.

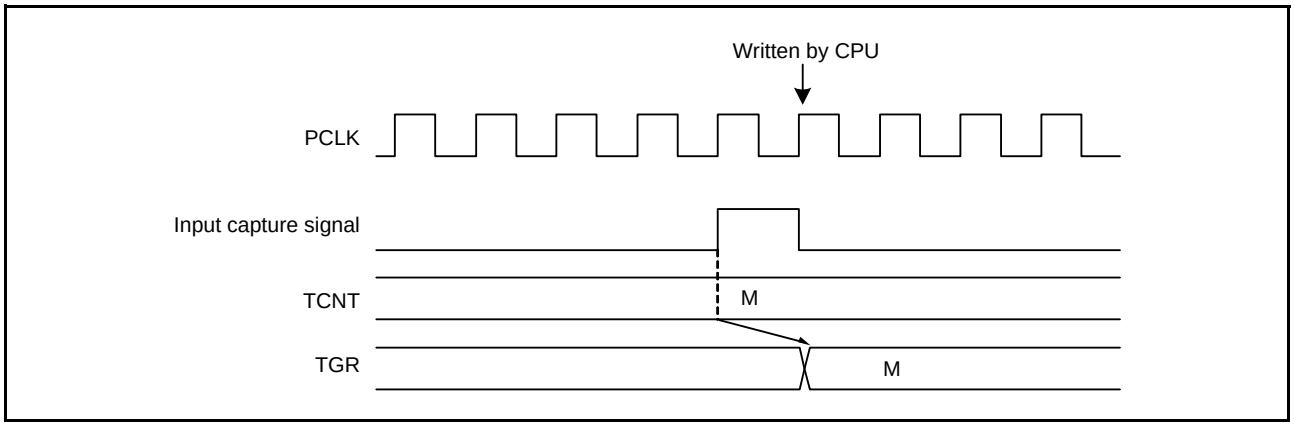


Figure 22.114 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)

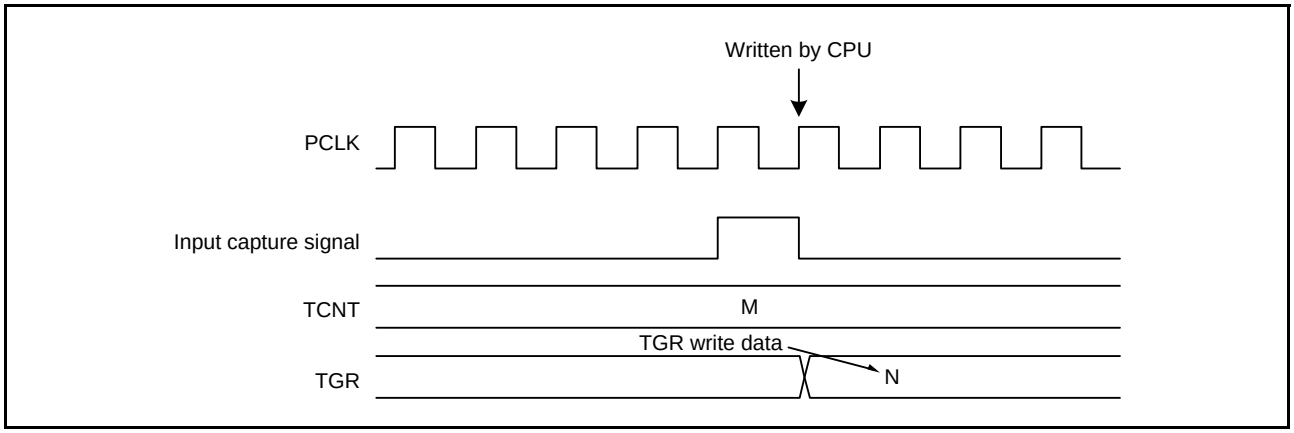


Figure 22.115 Contention between TGR Write Operation and Input Capture (MTU5)

22.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.
Figure 22.116 shows the timing in this case.

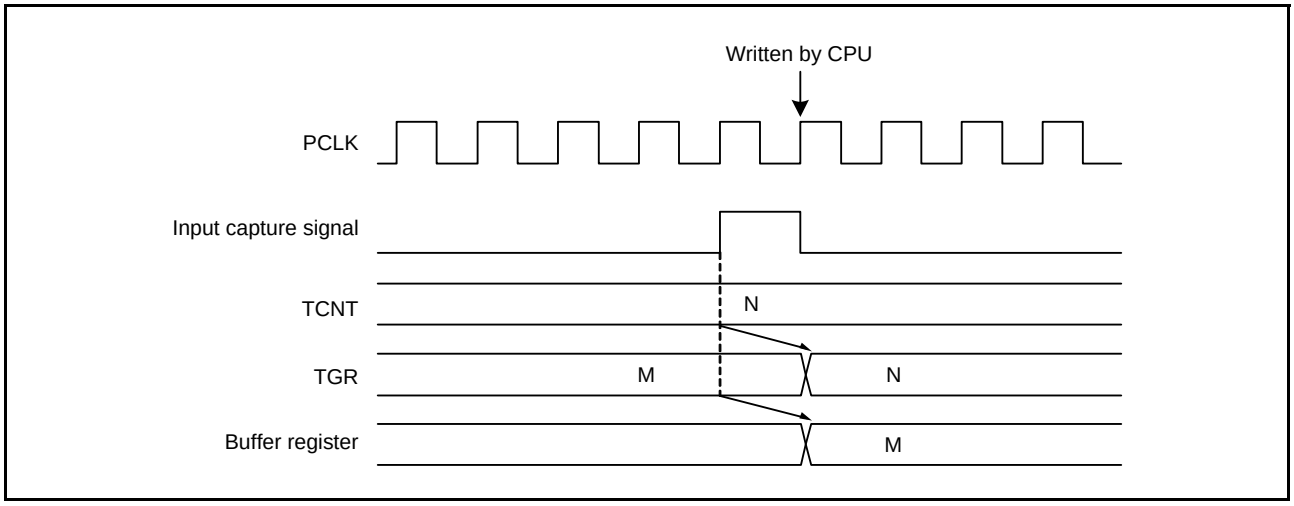


Figure 22.116 Contention between Buffer Register Write Operation and Input Capture

22.6.12 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 22.117 shows the timing in this case.
When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

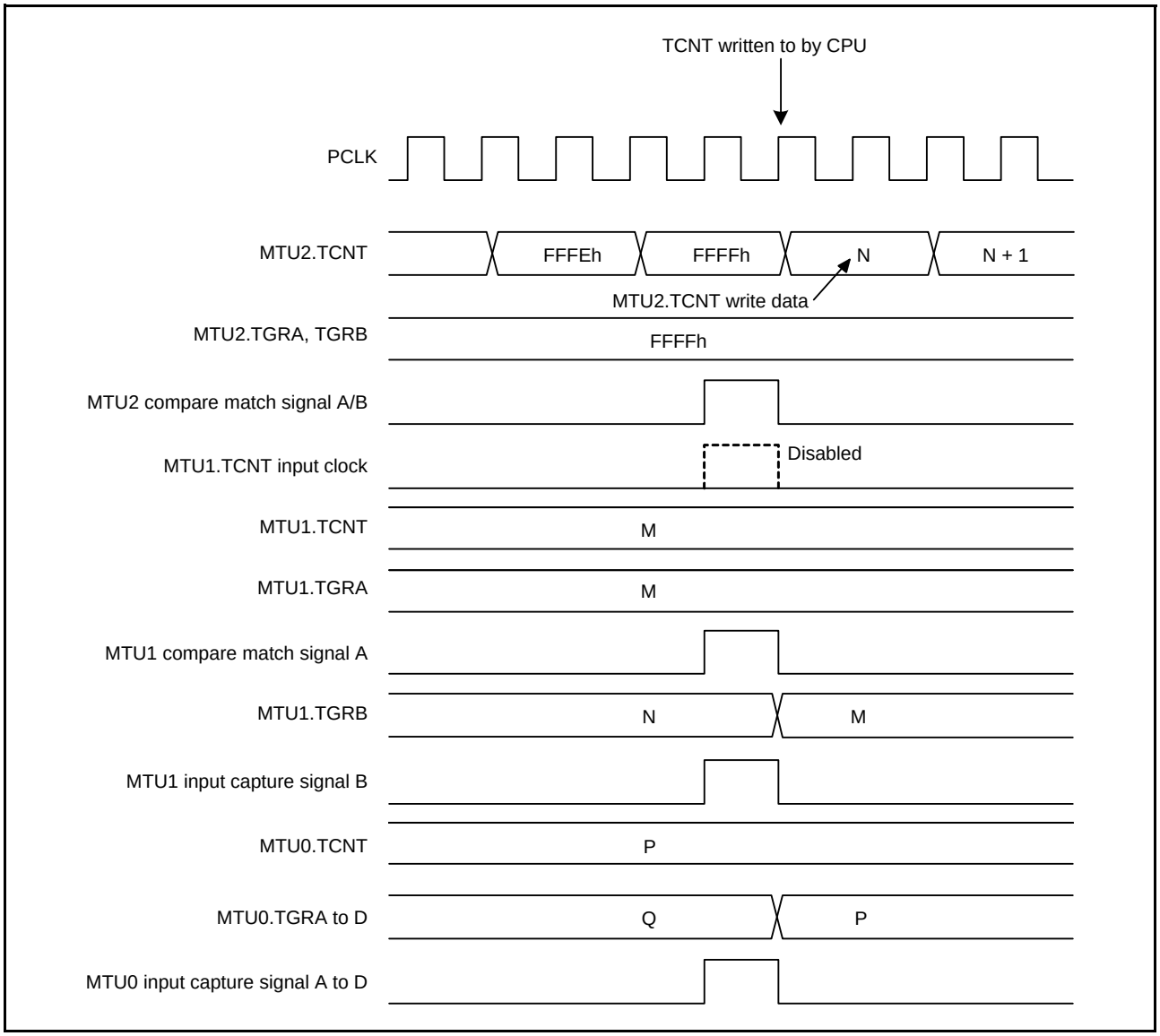


Figure 22.117 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

22.6.13 Counter Value when Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT is stopped in complementary PWM mode, MTU3.TCNT is set to the timer dead time register (TDDR) value and MTU4.TCNT is set to 0000h.

When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 22.118 shows this operation.

When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT.

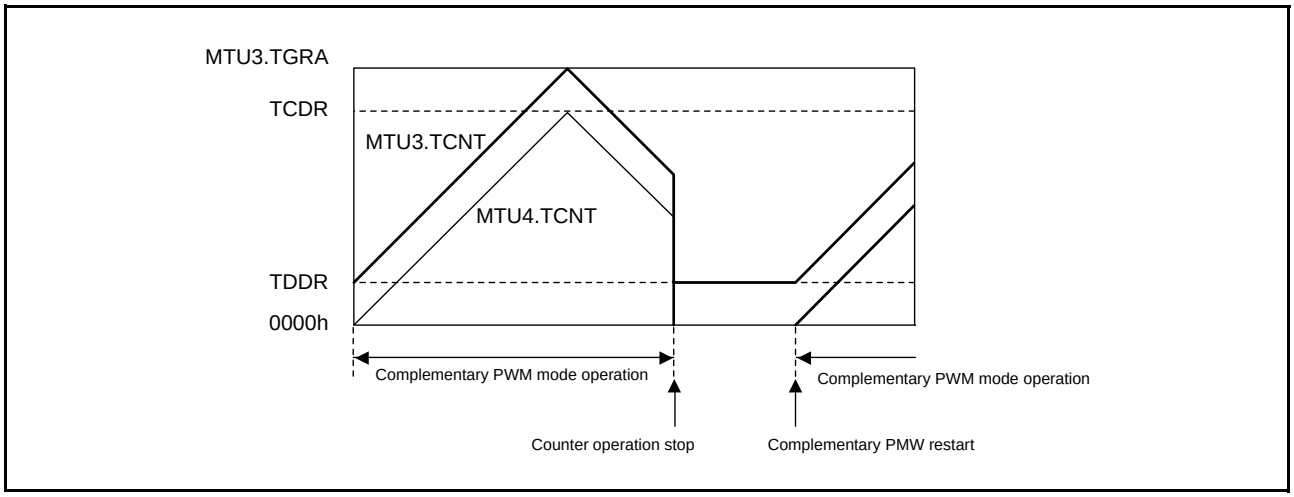


Figure 22.118 Counter Value when Stopped in Complementary PWM Mode (MTU3 and MTU4 Operation)

22.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTU3.TGRA), timer cycle data register (TCDR), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) in complementary PWM mode, be sure to use buffer operation. Also BFA and BFB bits in MTU4.TMDR should be set to 0. Setting the BFA bit in MTU4.TMDR to 1 disables MTIOC4C pin waveform output. Setting the BFB bit in MTU4.TMDR to 1 also disables MTIOC4D pin waveform output.
In complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in bits BFA and BFB of MTU3.TMDR. When the BFA bit in MTU3.TMDR is set to 1, MTU3.TGRC functions as a buffer register for MTU3.TGRA. At the same time, MTU4.TGRC functions as a buffer register for MTU4.TGRA, and TCBR functions as a buffer register for TCDR.

22.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR to 0. Setting the BFA bit in MTU4.TMDR to 1 disables MTIOC4C pin waveform output. Setting the BFB bit in MTU4.TMDR to 1 also disables MTIOC4D pin waveform output.
In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the BFA and BFB bits of MTU3.TMDR. For example, if the BFA bit in MTU3.TMDR is set to 1, MTU3.TGRC functions as a buffer register for MTU3.TGRA. At the same time, MTU4.TGRC functions as a buffer register for MTU4.TGRA.
While the MTU3.TGRC and MTU3.TGRD are operating as buffer registers, the corresponding TGIC and TGID interrupt requests are never generated.
Figure 22.119 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3m, and MTIOC4m operation with the BFA and BFB bits in MTU3.TMDR set to 1 and the BFA and BFB bits in MTU4.TMDR set to 0.

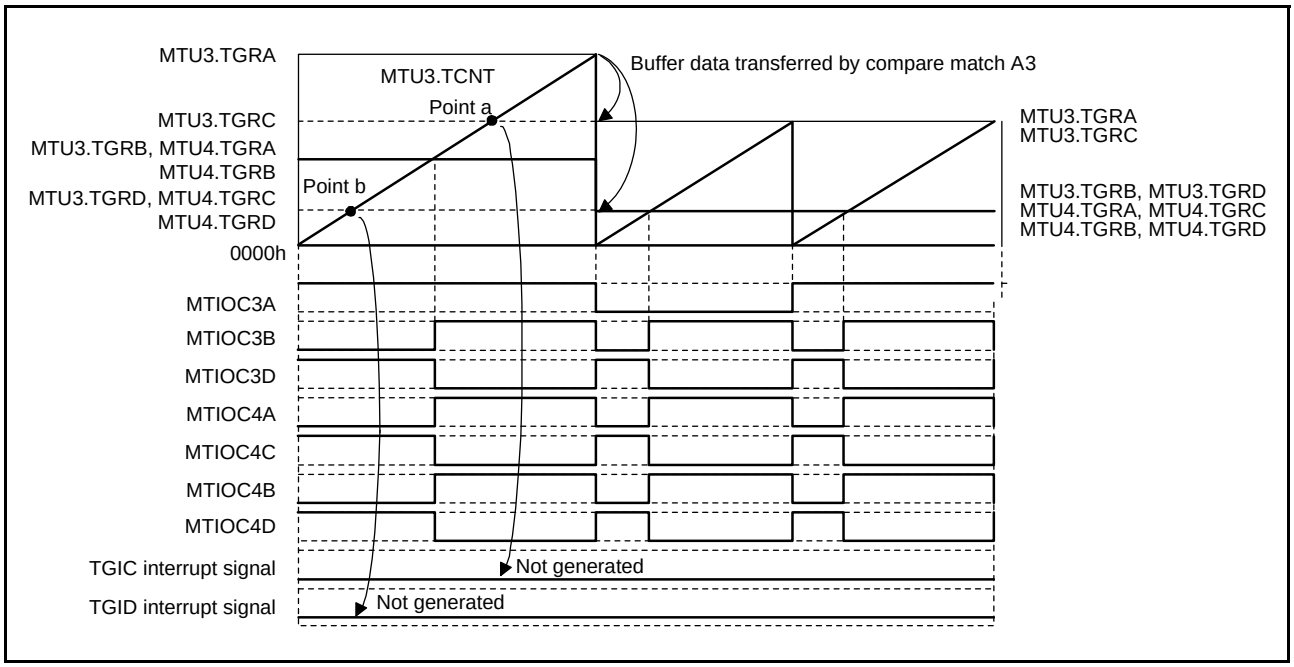


Figure 22.119 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

22.6.16 Overflow Flags in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT start counting when the CST3 bit of TSTR is set to 1. In this state, the MTU4.TCNT count clock source and count edge are determined by the MTU3.TCR setting.

In reset-synchronized PWM mode, with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT count up to FFFFh, then a compare match occurs with MTU3.TGRA, and MTU3.TCNT and MTU4.TCNT are both cleared. In this case, the corresponding TCIV interrupt request is not generated.

Figure 22.120 shows an operation example in reset-synchronized PWM mode with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match specified for the counter clearing source without synchronous operation setting.

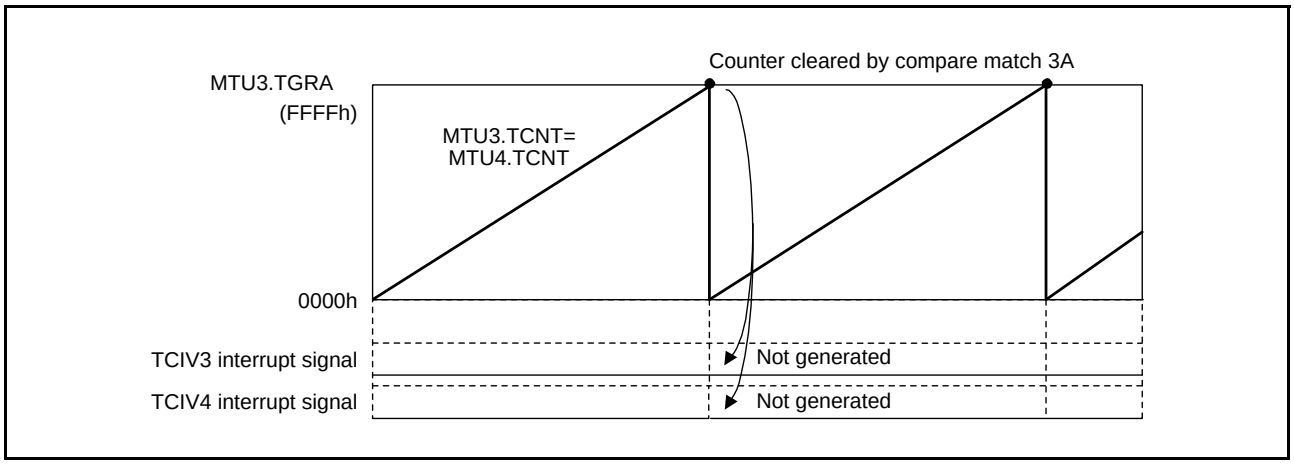


Figure 22.120 Overflow Flags in Reset-Synchronized PWM Mode

22.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, TCNT clearing takes precedence and the corresponding TCIV interrupt is not generated.
Figure 22.121 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

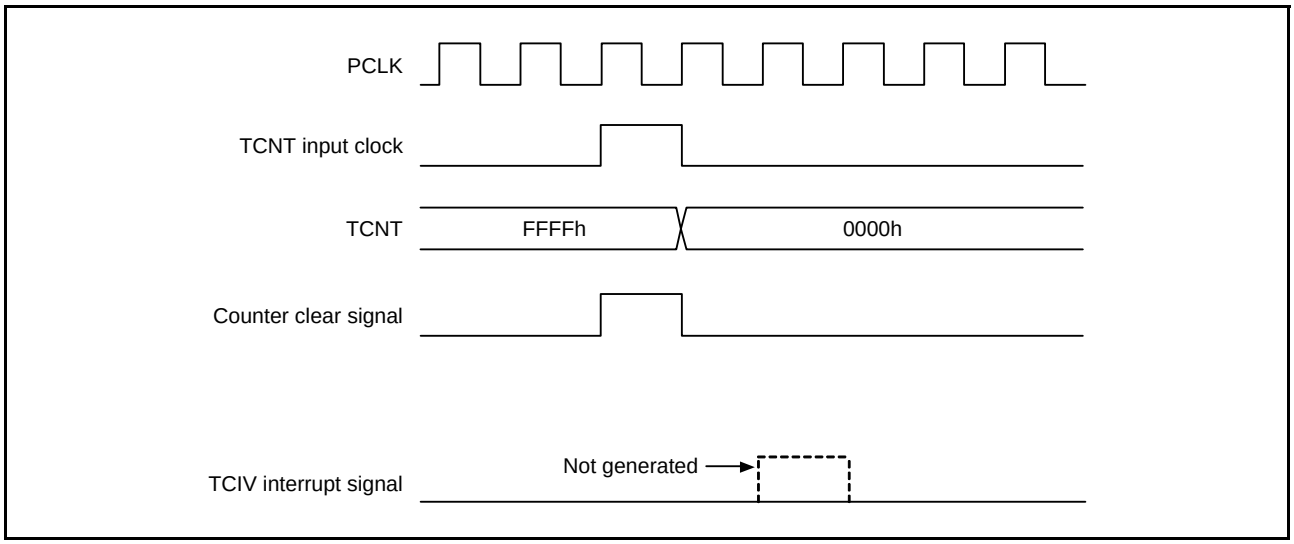


Figure 22.121 Contention between Overflow and Counter Clearing

22.6.18 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. The corresponding interrupt is not generated.
Figure 22.122 shows the operation timing when there is contention between TCNT write operation and overflow.

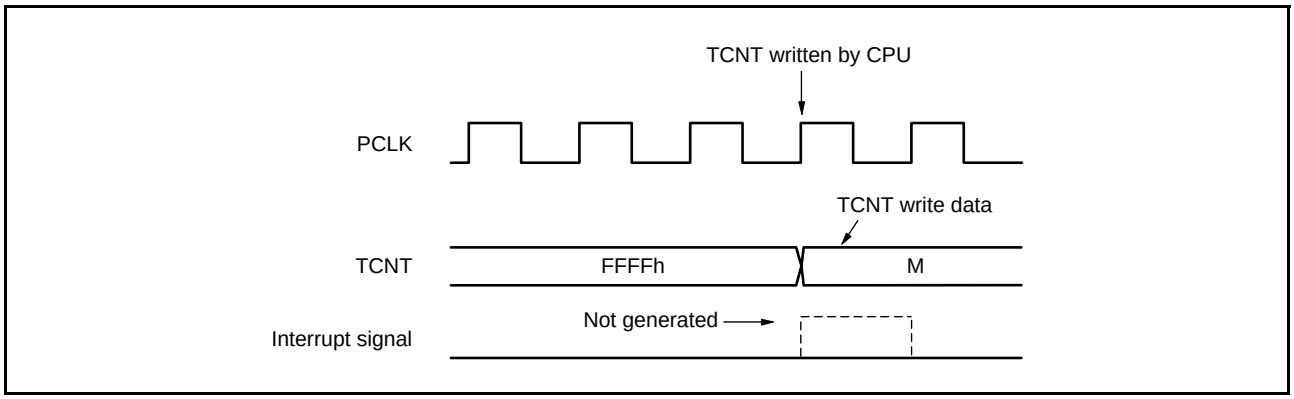


Figure 22.122 Contention between TCNT Write Operation and Overflow

22.6.19 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, if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D) 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 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.

22.6.20 Output Level in Complementary PWM Mode or Reset-Synchronized PWM Mode

When complementary PWM mode or reset-synchronized PWM mode is selected for MTU3 or MTU4, use the OLSP and OLSN bits in the timer output control register 1 (TOCR1) to set the levels for PWM waveform output. Also, when either of these modes is in use, set TIOR to 00h.

22.6.21 Interrupts During Periods in the Module-Stop State

When an module that has issued an interrupt request enters the module-stop state, clearing the source of the interrupt for the CPU or activation signal for the DMACA or DTC is not possible.
Accordingly, disable interrupts, etc. before making the settings for the module-stop state.

22.6.22 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 of 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 new function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input-capture input as the trigger. This function allows reading of the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, see section 22.2.8, Timer Input Capture Control Register (TICCR).

22.6.23 Notes when Complementary PWM Mode Output Protection Functions are not Used

The complementary PWM mode output protection functions are initially enabled. If the functions are not used, the POE.POECR2 register should be set to 00h.

22.6.24 Points for Caution to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCR.WRE = 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 PWM inverse-phase output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 22.123, 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 22.124, synchronous clearing occurs when any condition from among MTU3.TGRB ≤ TDDR, MTU4.TGRA ≤ TDDR, or MTU4.TGRB ≤ TDDR is satisfied.

The following method avoids the above phenomena.

- Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) set to at least double the value of the dead time data register (TDDR).

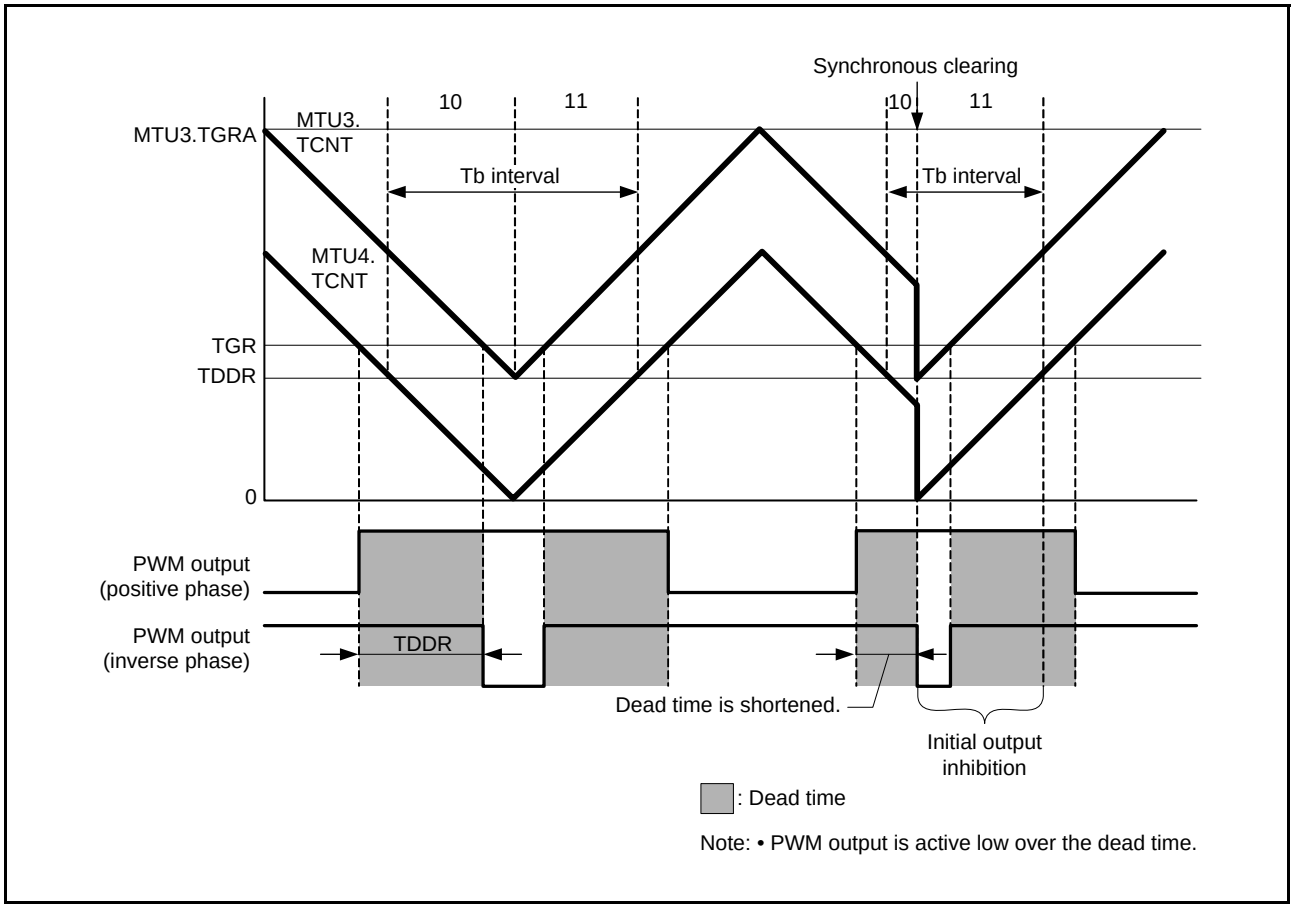


Figure 22.123 Example of Synchronous Clearing (When Condition 1 Applies)

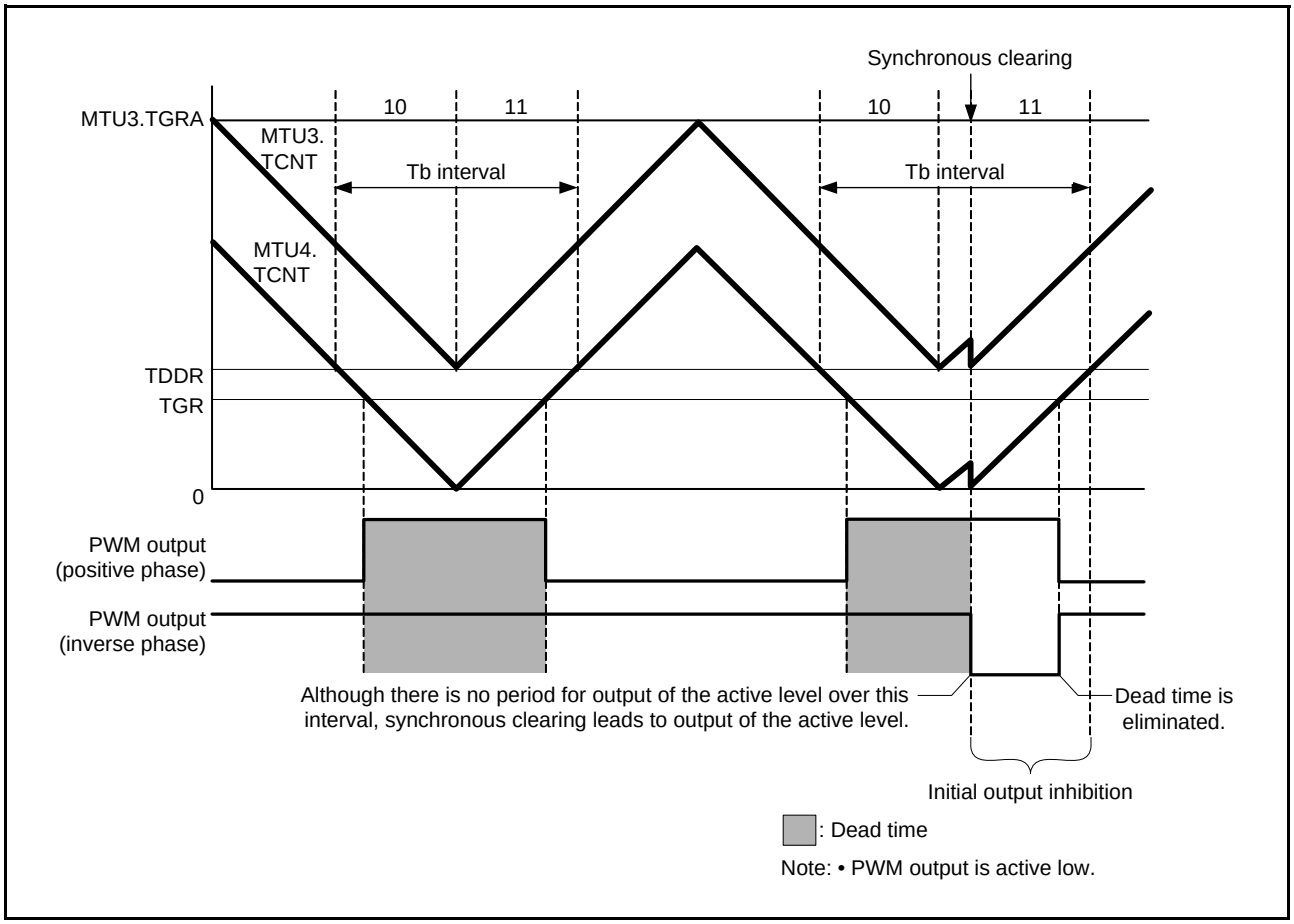


Figure 22.124 Example of Synchronous Clearing (When Condition 2 Applies)

22.6.25 Continuous Output of Compare-Match Pulse Interrupt Signal

When TGR is set to 0, PCLK/1 is set as the counter clock, and compare match is set as the counter clear source, the TCNT counter 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 third and subsequent interrupts.

Figure 21.125 shows an operation timing when the compare-match pulse interrupt signal is continuously output.

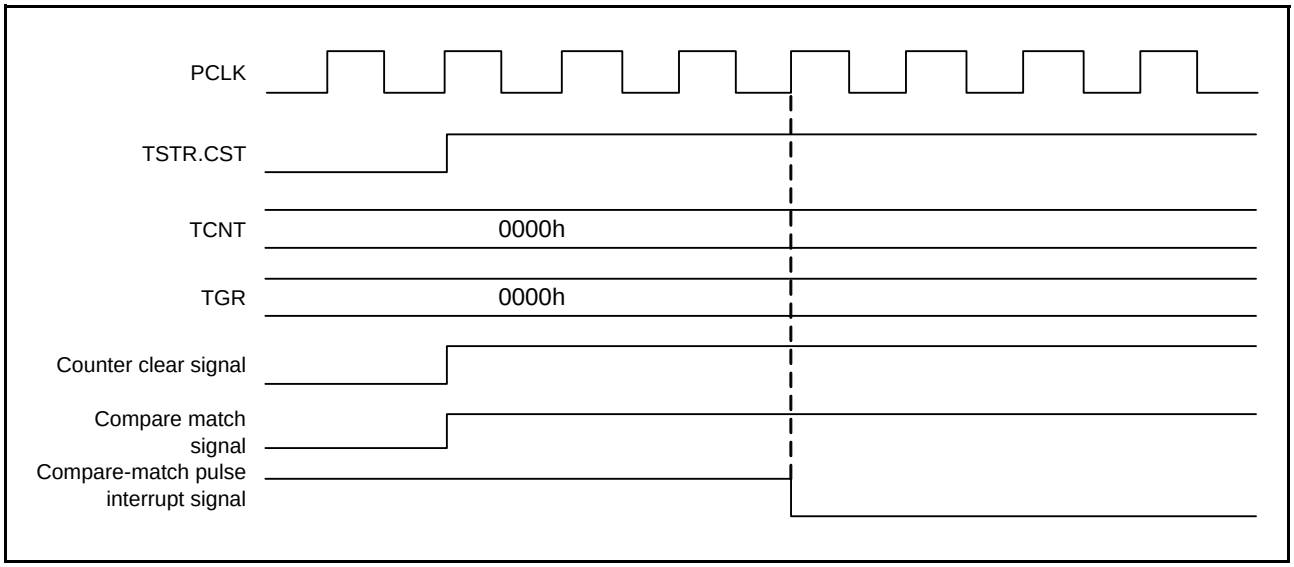


Figure 22.125 Continuous Output of Compare-Match Pulse Interrupt Signal

22.7 MTU Output Pin Initialization

22.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4)
- PWM mode 1 (MTU0 to MTU4)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 4 (MTU1 and MTU2)
- Complementary PWM mode (MTU3 and MTU4)
- Reset-synchronized PWM mode (MTU3 and MTU4)

This section describes how to initialize the MTU output pins in each of these modes.

22.7.2 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. Cutting MTU2A output off is achieved by switching the pins to the output port function and placing the inverse of the active level on the pins. For motor driving pins, output can also be cut by hardware, using port output enable 2 (POE). 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 listed in Table 22.59.

Table 22.59 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 4
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

22.7.3 Overview of Pin 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, since a waveform is not output to the MTIOCnB (MTIOCnD) pins, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pins, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, and temporarily disable output in MTU3 and MTU4 with the timer output master enable register (TOER). After that, operate the MTU2A in accordance with the mode setting procedure (TOCR setting, TMDR setting, and TOER setting).

Note: • Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in Table 22.59. The active level is assumed to be low.

(1) Operation when Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 22.126 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

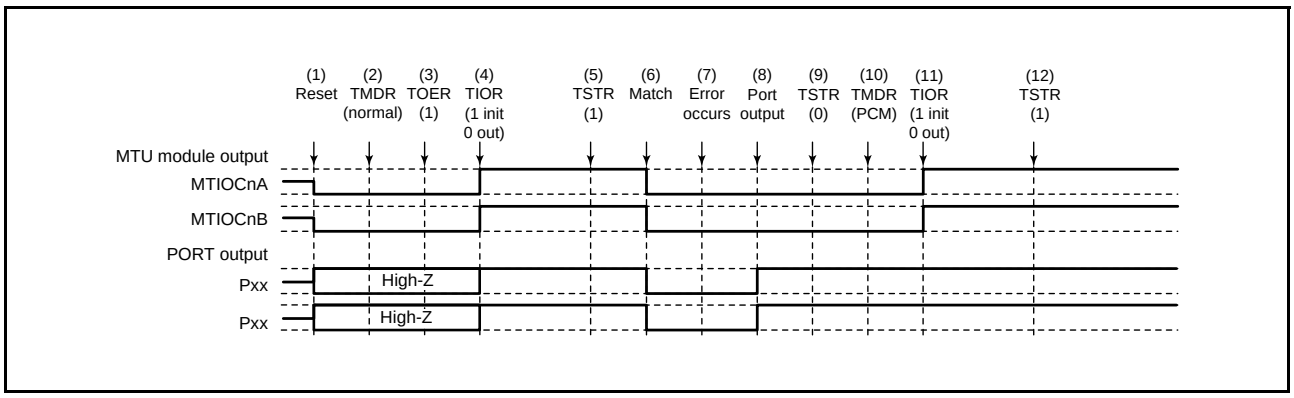


Figure 22.126 Error Occurrence in Normal Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) After a reset, the TMDR setting is for normal mode.
(3) For MTU3 and MTU4, enable output with TOER before initializing the pins with TIOR.
(4) Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(5) Start count operation by setting TSTR.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Disable the pin output and select port output with TIOR and output the inverse of the active level.
(9) Stop count operation by setting TSTR.
(10) This step is not necessary when restarting in normal mode.
(11) Initialize the pins with TIOR.
(12) Restart operation by setting TSTR.

(2) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 22.127 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

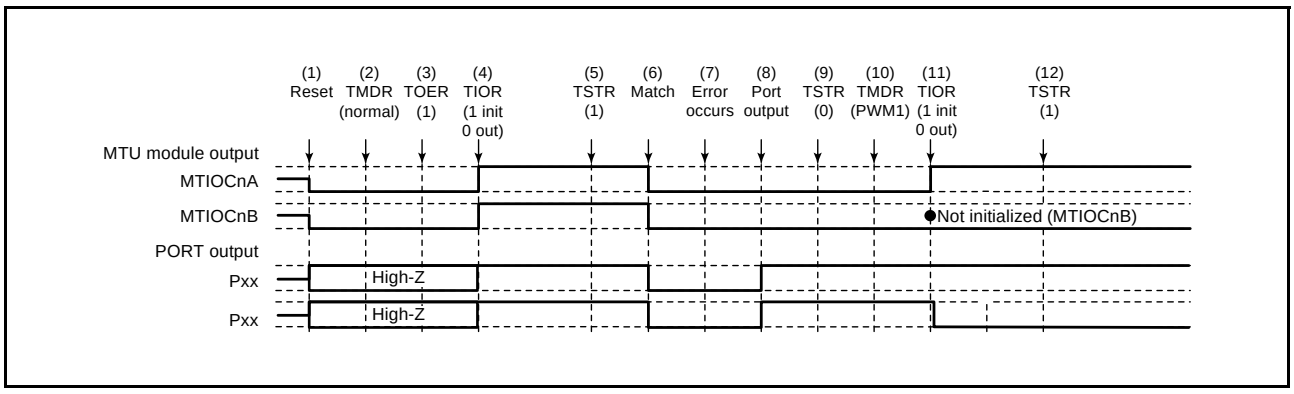


Figure 22.127 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 22.126.
(10) Set PWM mode 1.
(11) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 1.)
(12) Restart operation by setting TSTR.

(3) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 22.128 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

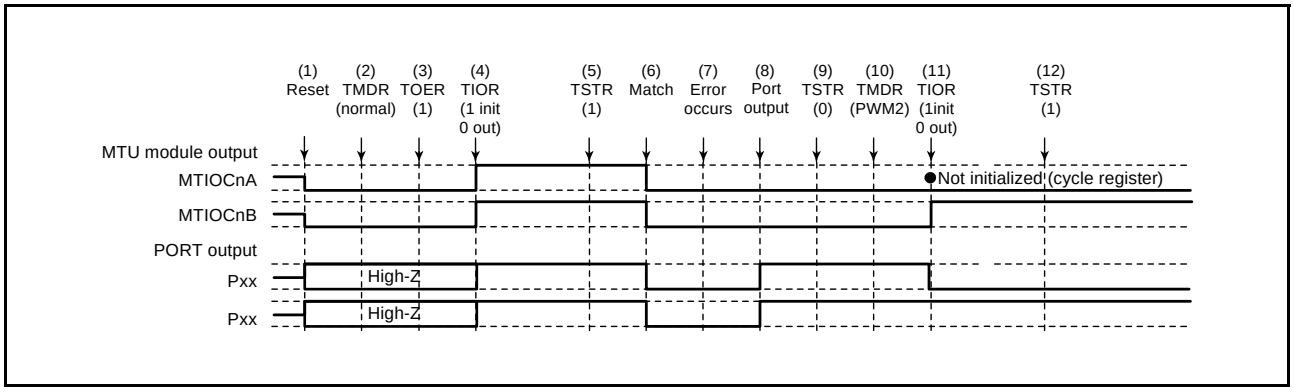


Figure 22.128 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 22.126.
(10) Set PWM mode 2.
(11) Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)
(12) Restart operation by setting TSTR.

Note: • PWM mode 2 can only be selected for MTU0 to MTU2, and therefore TOER setting is not necessary.

(4) Operation when Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 22.129 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

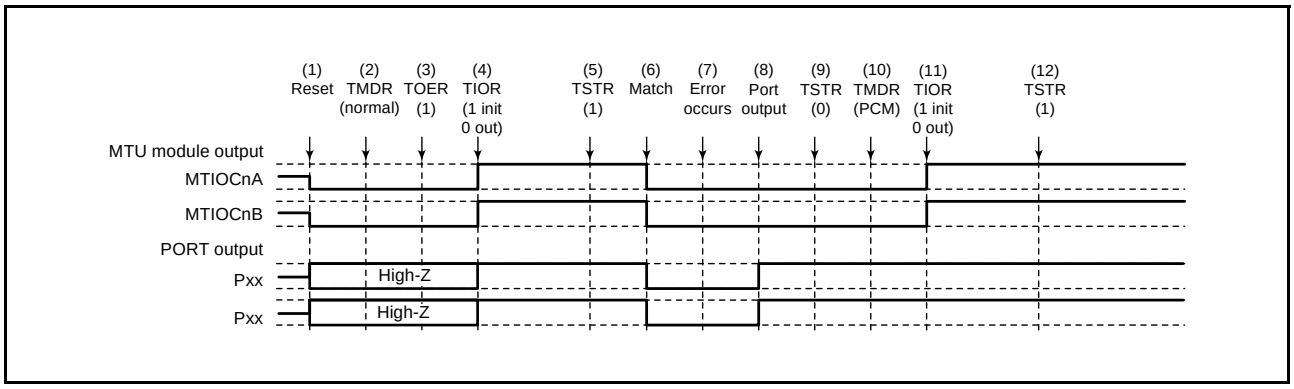


Figure 22.129 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 22.126.
(10) Set the phase counting mode.
(11) Initialize the pins with TIOR.
(12) Restart operation by setting TSTR.

Note: • The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOER setting is not necessary.

(5) Operation when Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.130 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

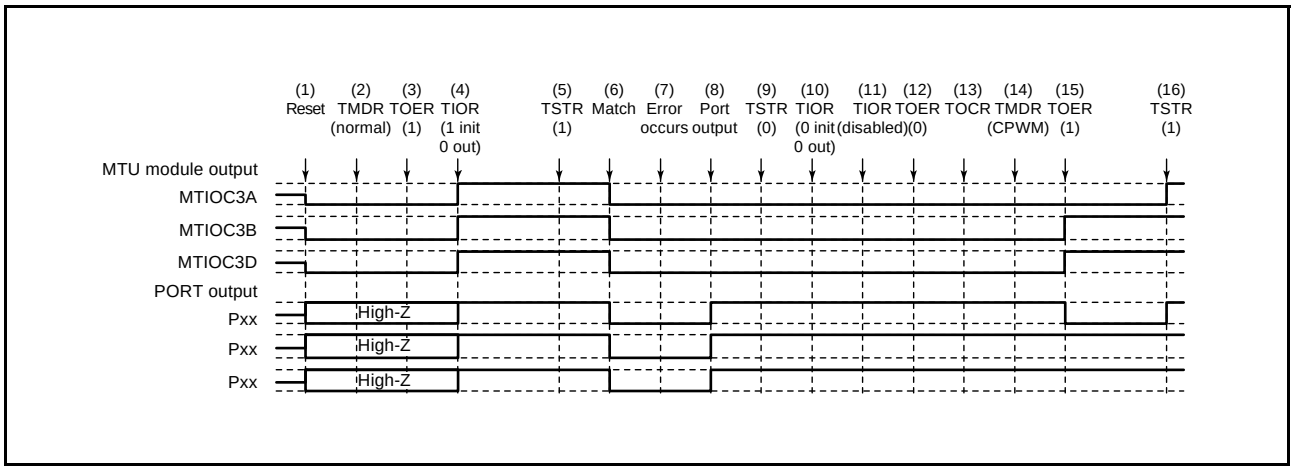


Figure 22.130 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 22.126.
(10) Initialize the normal mode waveform generation section with TIOR.
(11) Disable operation of the normal mode waveform generation section with TIOR.
(12) Disable output in MTU3 and MTU4 with TOER.
(13) Select the complementary PWM output level and enable or disable cyclic output with TOCR.
(14) Set complementary PWM mode.
(15) Enable output in MTU3 and MTU4 with TOER.
(16) Restart operation by setting TSTR.

(6) Operation when Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.131 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

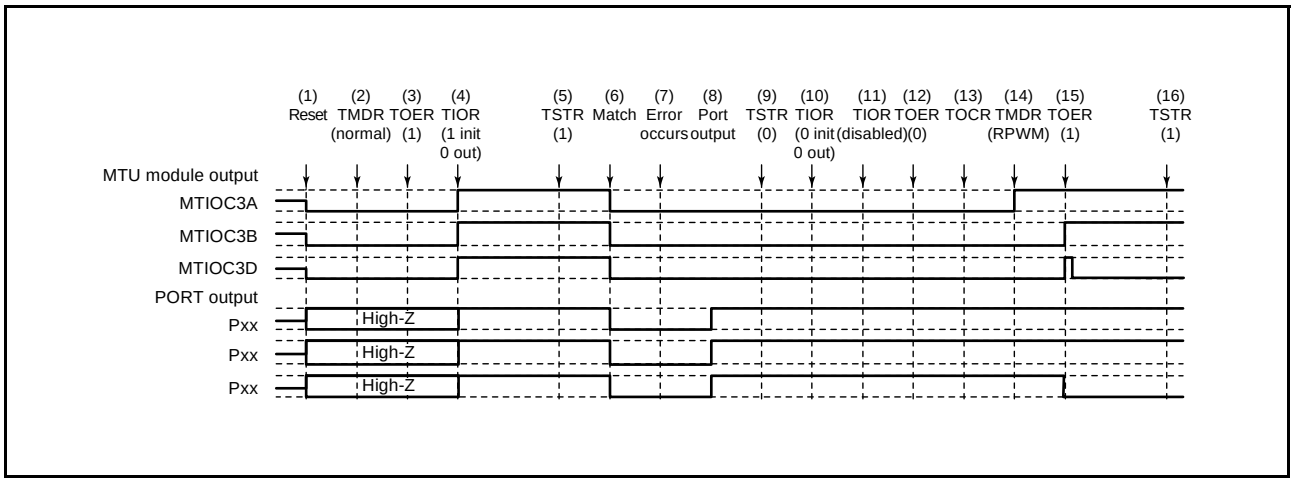


Figure 22.131 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (12) are the same as in Figure 22.126.
(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.
(14) Set reset-synchronized PWM mode.
(15) Enable output in MTU3 and MTU4 with TOER.
(16) Restart operation by setting TSTR.

(7) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 22.132 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

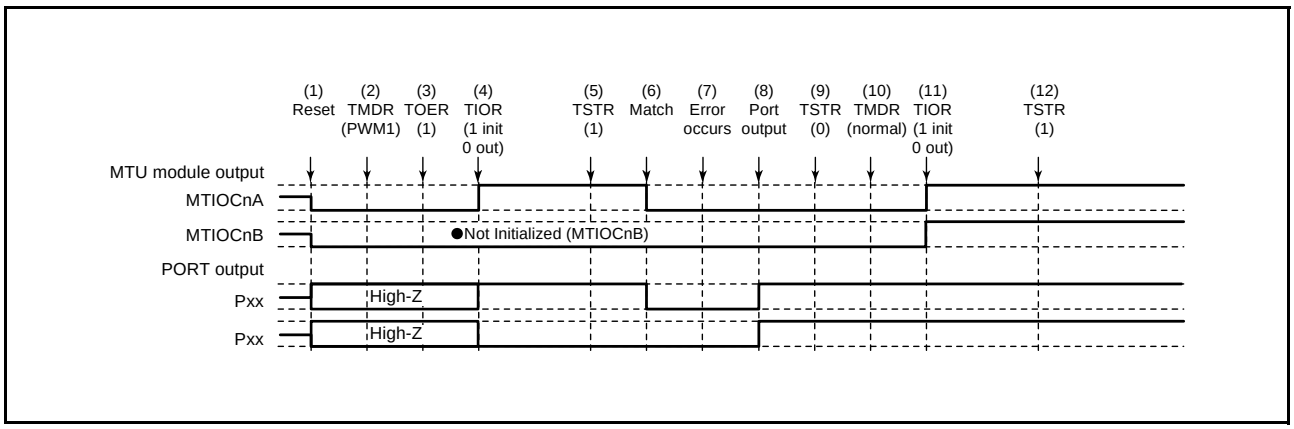


Figure 22.132 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(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 TOER before initializing the pins with TIOR.
(4) Initialize the pins with TIOR. (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) Start count operation by setting TSTR.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Disable the pin output and select port output with TIOR and output the inverse of the active level.
(9) Stop count operation by setting TSTR.
(10) Set normal mode.
(11) Initialize the pins with TIOR.
(12) Restart operation by setting TSTR.

(8) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 22.133 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

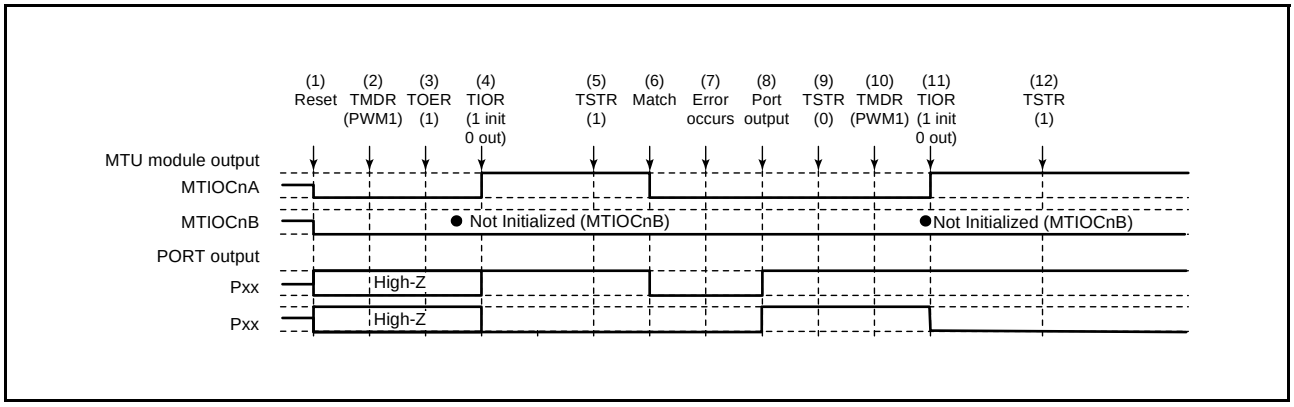


Figure 22.133 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 22.132.
(10) This step is not necessary when restarting in PWM mode 1.
(11) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)
(12) Restart operation by setting TSTR.

(9) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 22.134 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

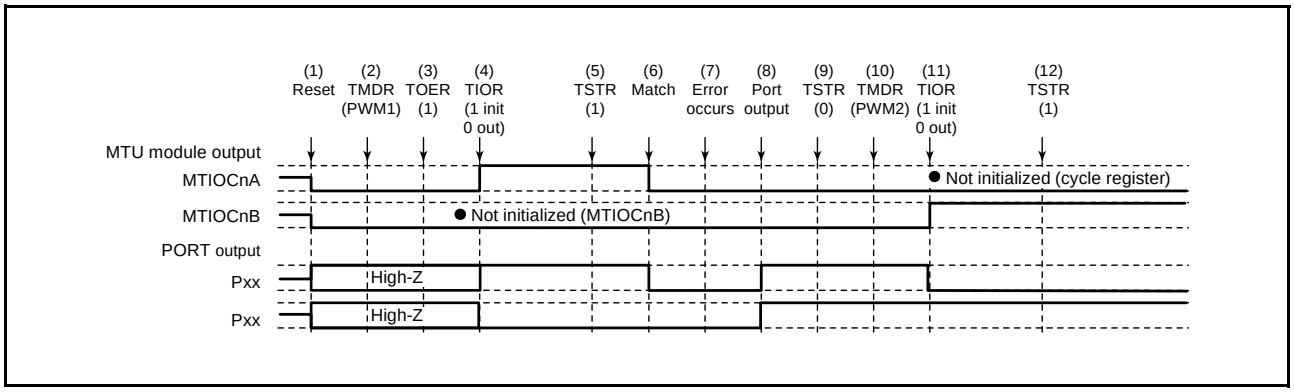


Figure 22.134 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 22.132.
(10) Set PWM mode 2.
(11) Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
(12) Restart operation by setting TSTR.
Note: • PWM mode 2 can only be selected for MTU0 to MTU2, and therefore, TOER setting is not necessary.

(10) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 22.135 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

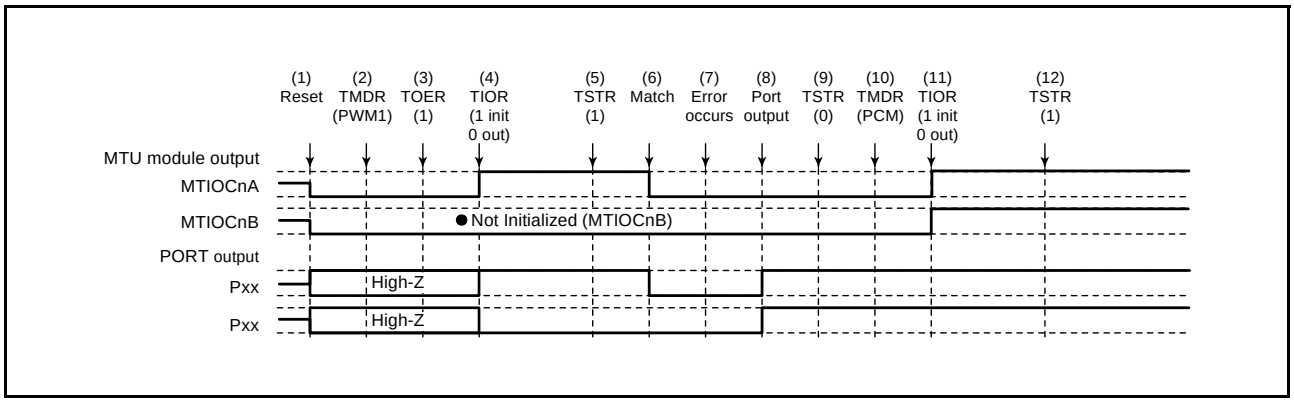


Figure 22.135 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 22.132.
(10) Set the phase counting mode.
(11) Initialize the pins with TIOR.
(12) Restart operation by setting TSTR.

Note: • The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOER setting is not necessary.

(11) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 22.136 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

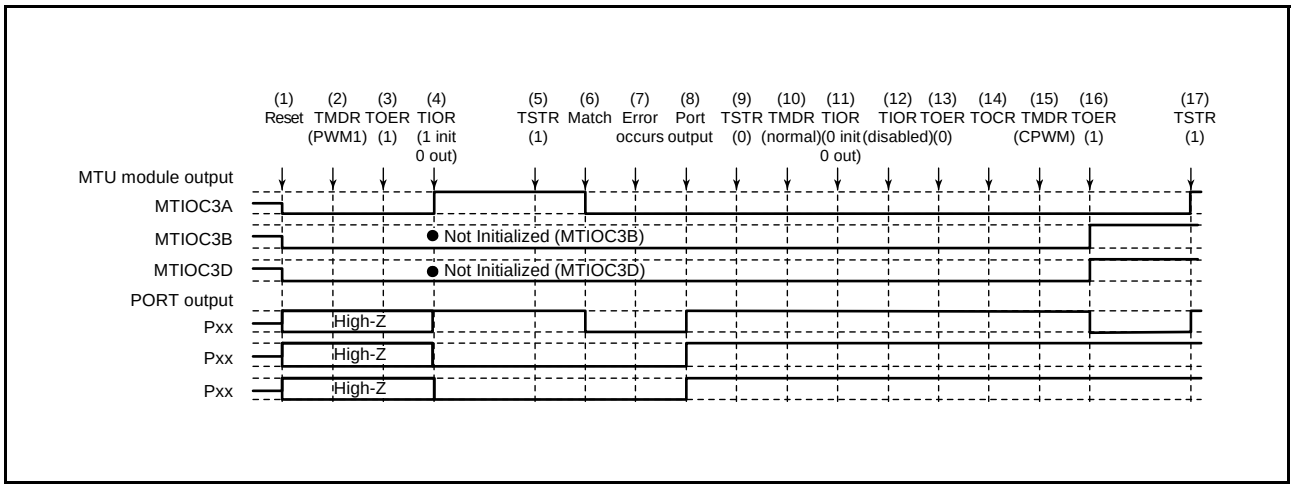


Figure 22.136 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 22.132.
(10) Set normal mode to initialize the normal mode waveform generation section.
(11) Initialize the PWM mode 1 waveform generation section with TIOR.
(12) Disable operation of the PWM mode 1 waveform generation section with TIOR
(13) Disable output in MTU3 and MTU4 with TOER.
(14) Select the complementary PWM output level and enable or disable cyclic output with TOCR.
(15) Set complementary PWM mode.
(16) Enable output in MTU3 and MTU4 with TOER.
(17) Restart operation by setting TSTR.

(12) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.137 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

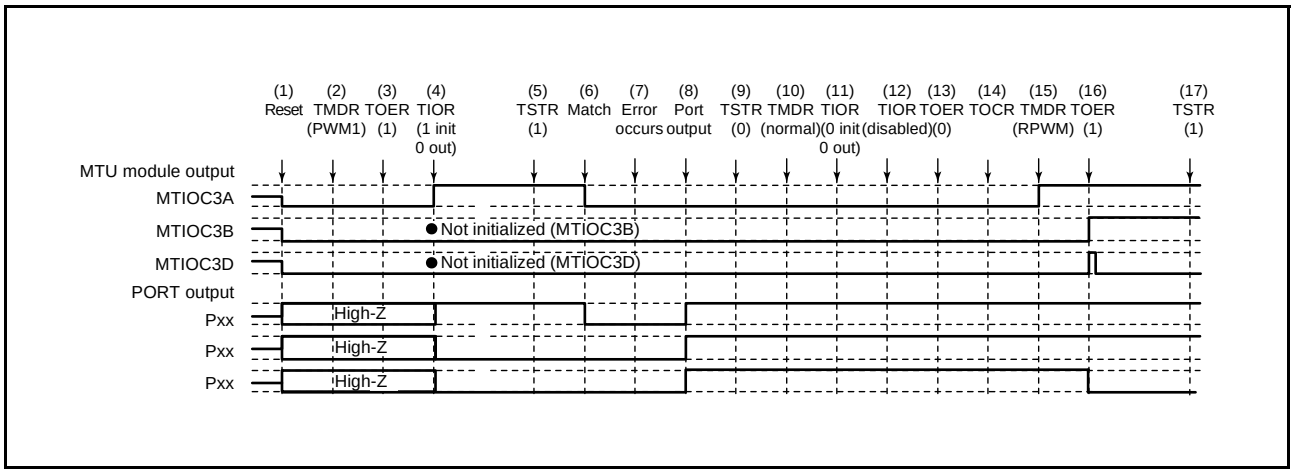


Figure 22.137 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 22.136.
(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.
(15) Set reset-synchronized PWM mode.
(16) Enable output in MTU3 and MTU4 with TOER.
(17) Restart operation by setting TSTR.

(13) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 22.138 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

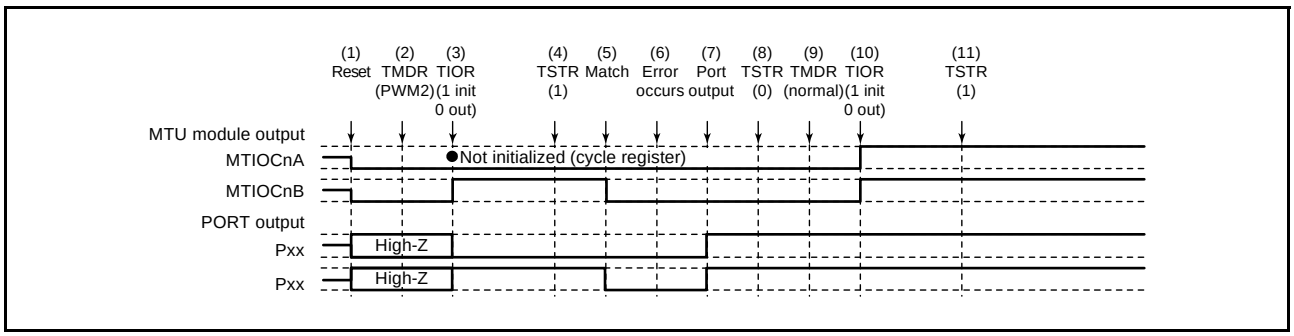


Figure 22.138 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(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 TIOR. (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, MTIOCnA is the cycle register.)
(4) Start count operation by setting TSTR.
(5) Output goes low on compare match occurrence.
(6) An error occurs.
(7) Disable the pin output and select port output with TIOR and output the inverse of the active level.
(8) Stop count operation by setting TSTR.
(9) Set normal mode.
(10) Initialize the pins with TIOR.
(11) Restart operation by setting TSTR.

(14) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 22.139 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

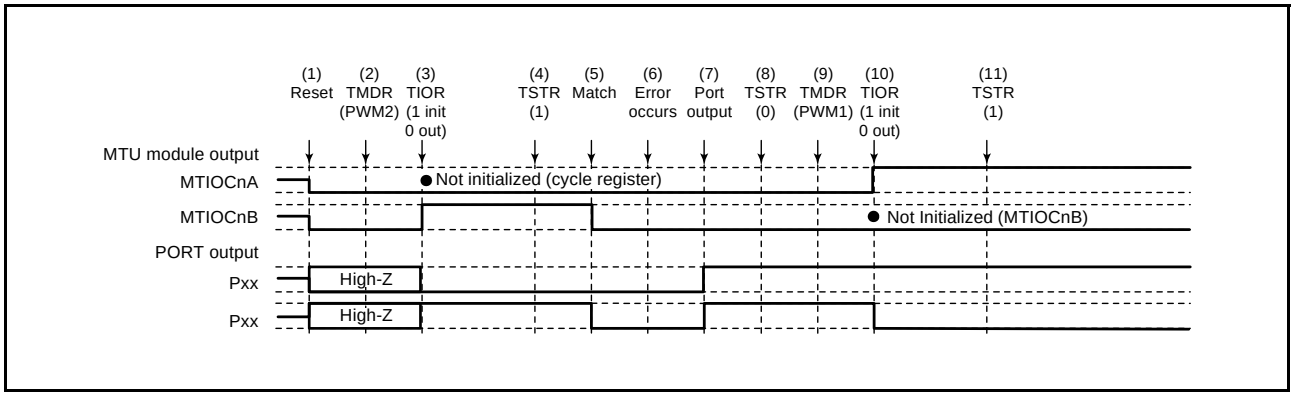


Figure 22.139 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (8) are the same as in Figure 22.138.
(9) Set PWM mode 1.
(10) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)
(11) Restart operation by setting TSTR.

(15) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 22.140 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

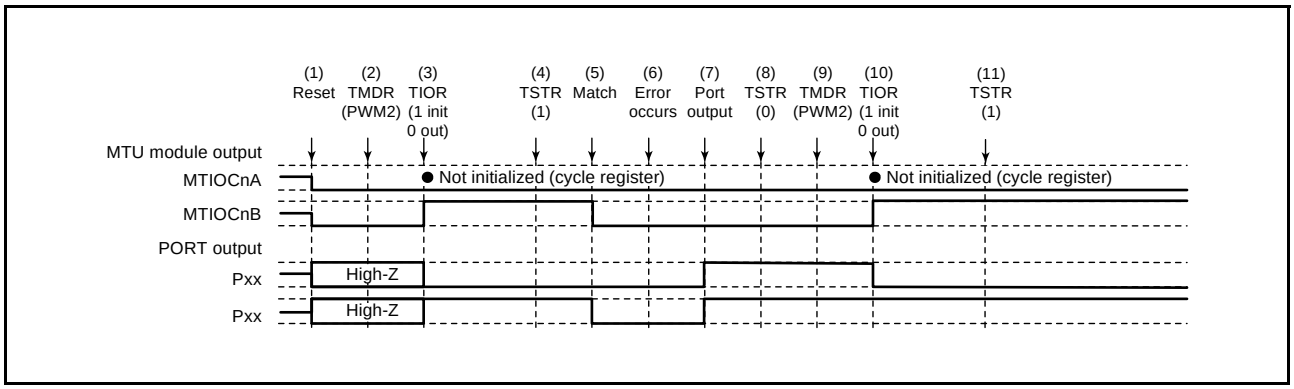


Figure 22.140 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (8) are the same as in Figure 22.138.
(9) This step is not necessary when restarting in PWM mode 2.
(10) Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
(11) Restart operation by setting TSTR.

(16) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 22.141 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

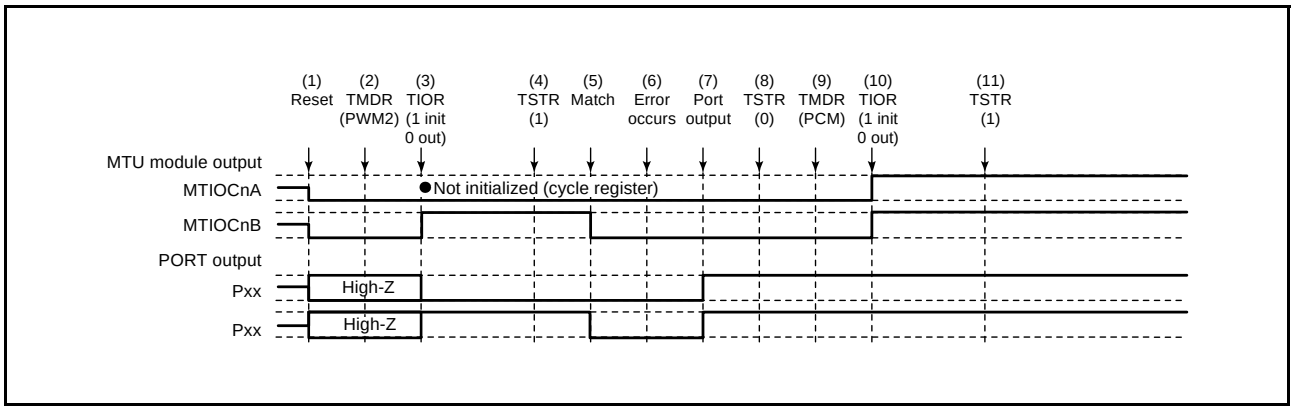


Figure 22.141 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (8) are the same as in Figure 22.138.
(9) Set the phase counting mode.
(10) Initialize the pins with TIOR.
(11) Restart operation by setting TSTR.

(17) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 22.142 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

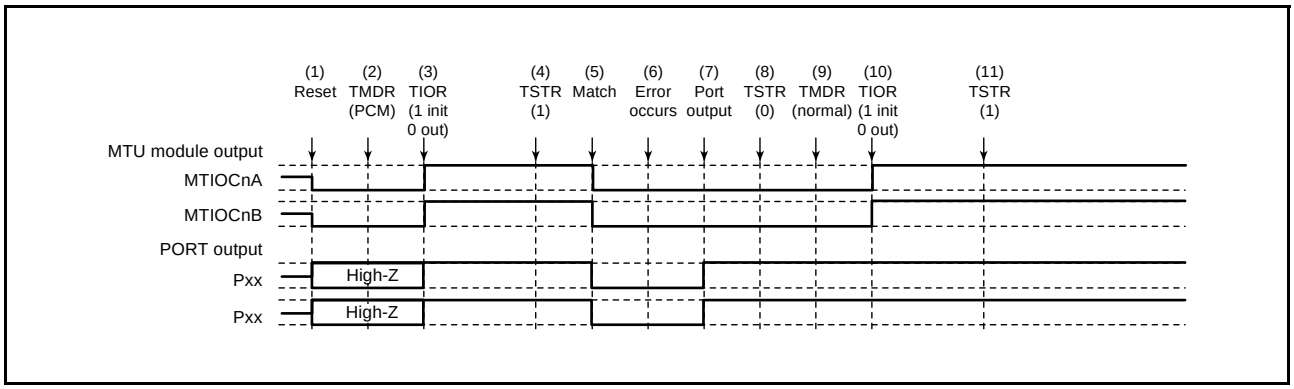


Figure 22.142 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(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 TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(4) Start count operation by setting TSTR.
(5) Output goes low on compare match occurrence.
(6) An error occurs.
(7) Select port output and output the inverse of the active level.
(8) Stop count operation by setting TSTR.
(9) Set normal mode.
(10) Initialize the pins with TIOR.
(11) Restart operation by setting TSTR.

(18) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 22.143 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

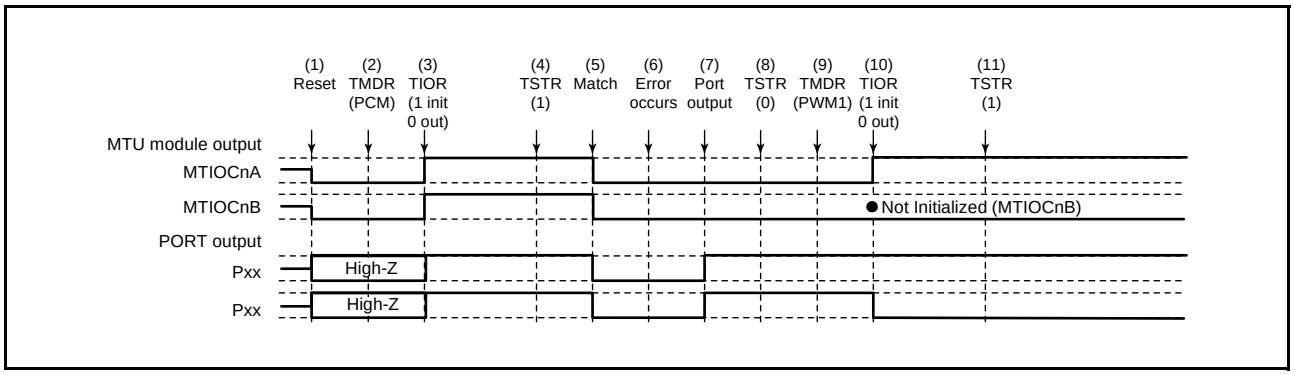


Figure 22.143 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (8) are the same as in Figure 22.142.
(9) Set PWM mode 1.
(10) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)
(11) Restart operation by setting TSTR.

(19) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 22.144 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

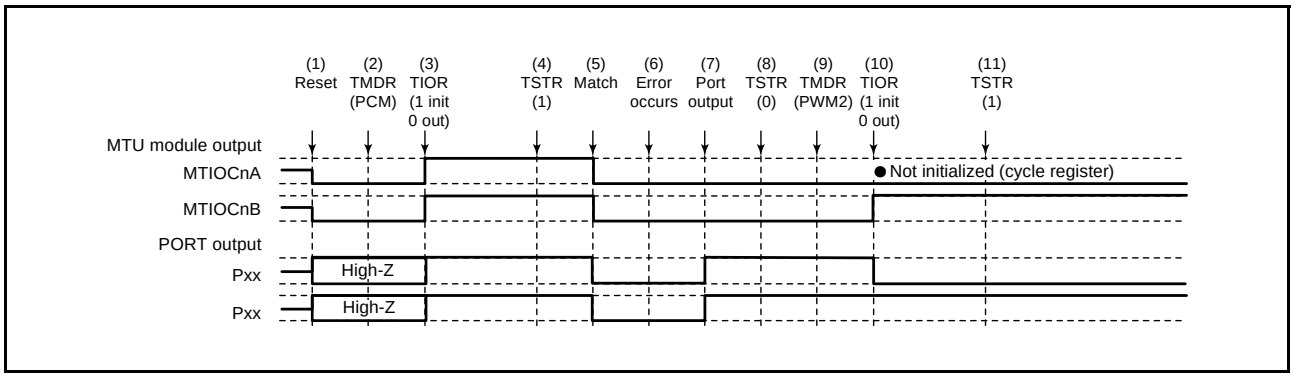


Figure 22.144 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (8) are the same as in Figure 22.142.
(9) Set PWM mode 2.
(10) Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
(11) Restart operation by setting TSTR.

(20) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 22.145 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

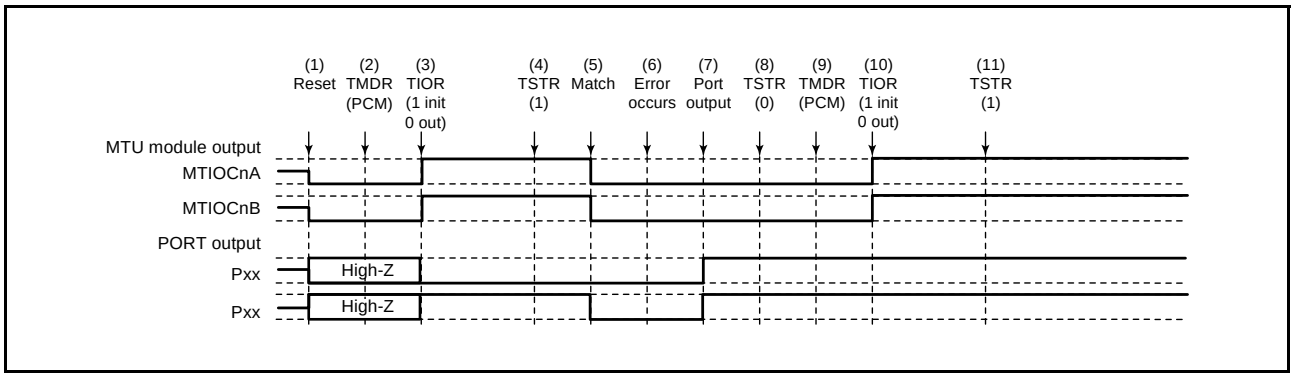


Figure 22.145 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (8) are the same as in Figure 22.142.
(9) This step is not necessary when restarting in phase counting mode.
(10) Initialize the pins with TIOR.
(11) Restart operation by setting TSTR.

(21) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 22.146 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

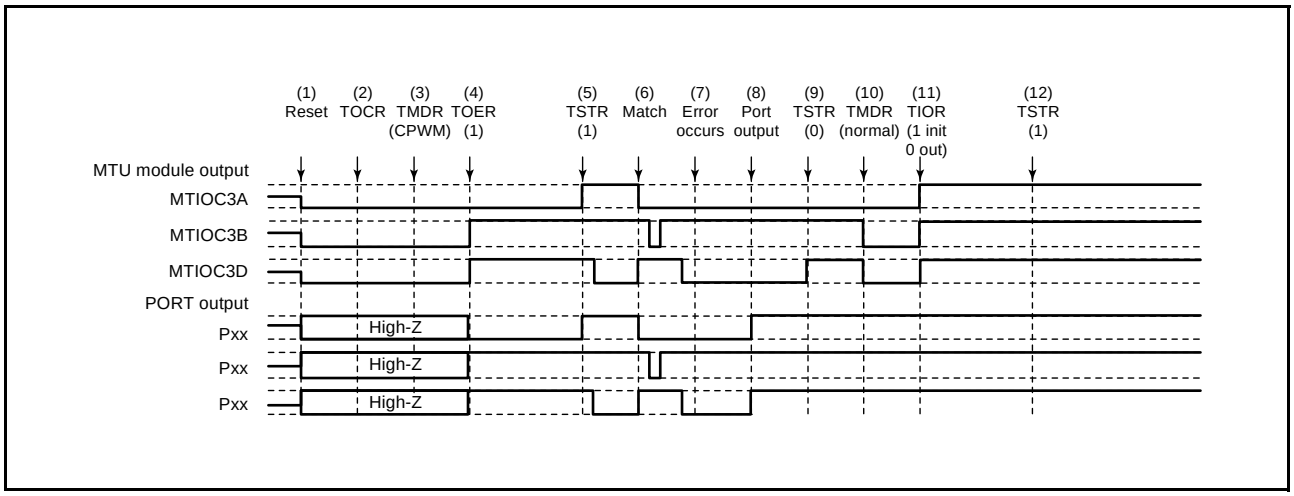


Figure 22.146 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Select the complementary PWM output level and enable or disable cyclic output with TOCR.
(3) Set complementary PWM mode.
(4) Enable output in MTU3 and MTU4 with TOER.
(5) Start count operation by setting TSTR.
(6) The complementary PWM waveform is output on compare match occurrence.
(7) An error occurs.
(8) Disable the pin output and select port output with TIOR and output the inverse of the active level.
(9) Stop count operation by setting TSTR. (MTU output becomes the initial complementary PWM output value).
(10) Set normal mode (MTU output goes low).
(11) Initialize the pins with TIOR.
(12) Restart operation by setting TSTR.

(22) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 22.147 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

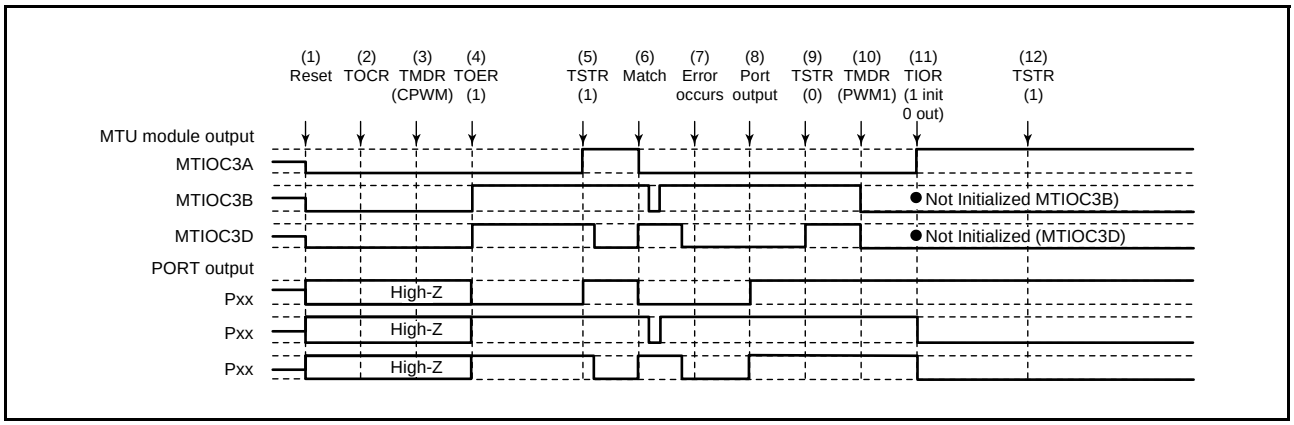


Figure 22.147 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 22.146.
(10) Set PWM mode 1 (MTU output goes low).
(11) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)
(12) Restart operation by setting TSTR.

(23) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.148 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).

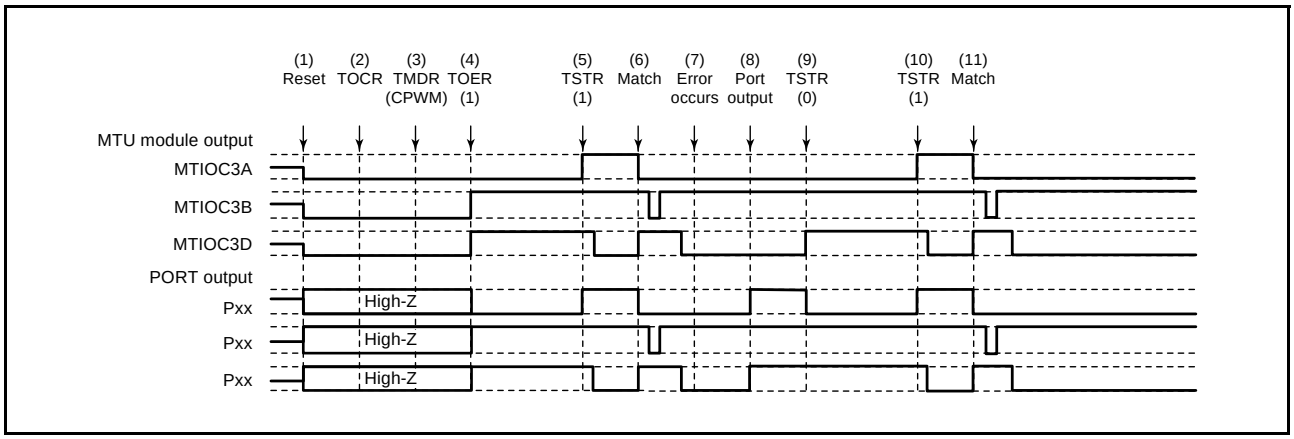


Figure 22.148 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 22.146.
(10) Restart operation by setting TSTR.
(11) 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 22.149 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 settings).

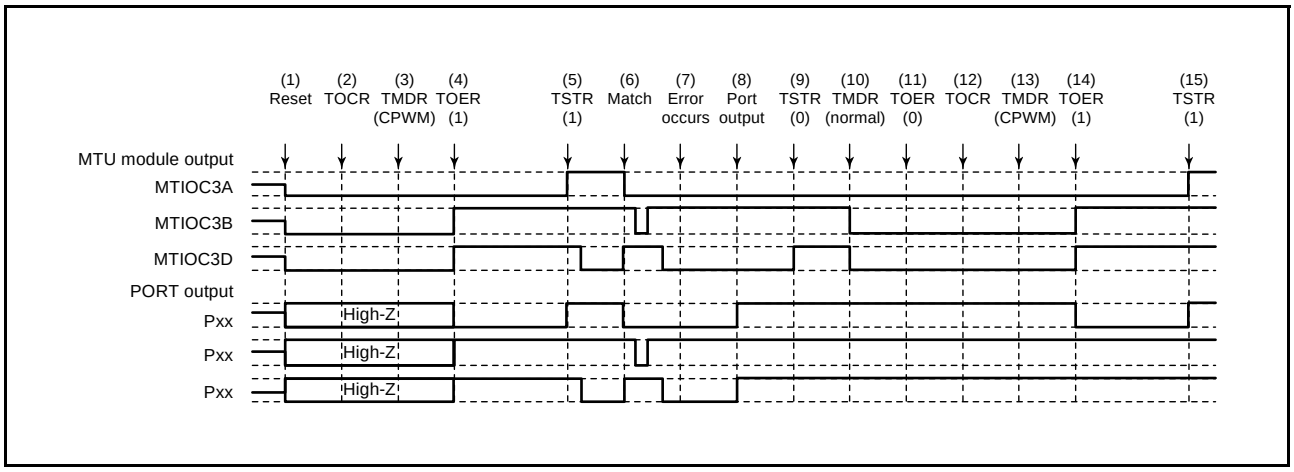


Figure 22.149 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 22.146.
(10) Set normal mode and make new settings (MTU output goes low).
(11) Disable output in MTU3 and MTU4 with TOER.
(12) Select the complementary PWM output level and enable or disable cyclic output with TOCR.
(13) Set complementary PWM mode.
(14) Enable output in MTU3 and MTU4 with TOER.
(15) Restart operation by setting TSTR.

(25) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.150 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

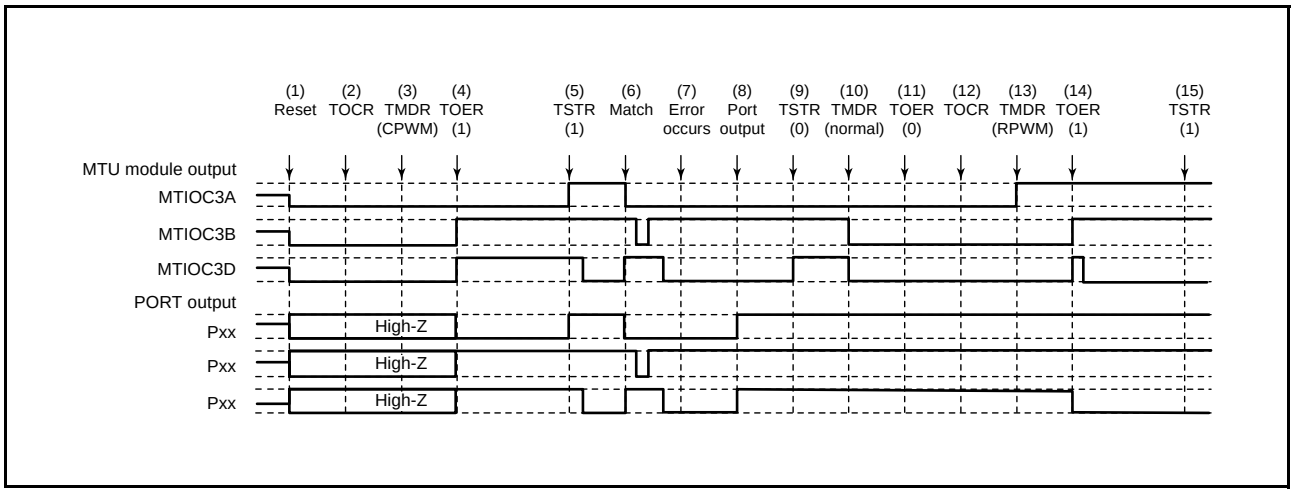


Figure 22.150 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (9) are the same as in Figure 22.146.
(10) Set normal mode (MTU output goes low).
(11) Disable output in MTU3 and MTU4 with TOER.
(12) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.
(13) Set reset-synchronized PWM mode.
(14) Enable output in MTU3 and MTU4 with TOER.
(15) Restart operation by setting TSTR.

(26) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 22.151 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

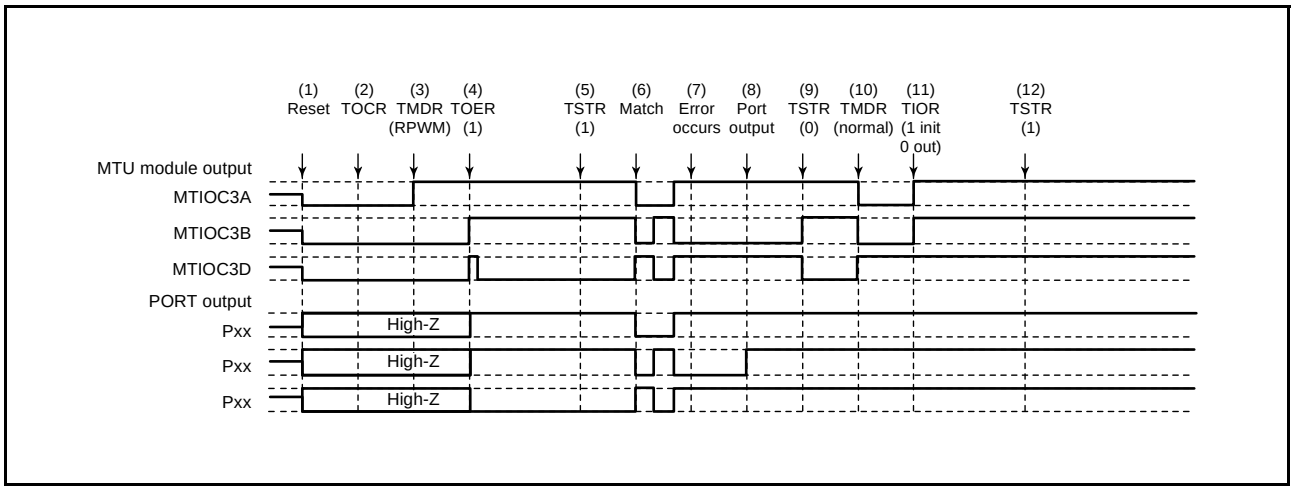


Figure 22.151 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(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 TOCR.
(3) Set reset-synchronized PWM mode.
(4) Enable output in MTU3 and MTU4 with TOER.
(5) Start count operation by setting TSTR.
(6) The reset-synchronized PWM waveform is output on compare match occurrence.
(7) An error occurs.
(8) Disable the pin output and select port output with TIOR and output the inverse of the active level.
(9) Stop count operation by setting TSTR. (MTU output becomes the initial reset-synchronized PWM output value.)
(10) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).
(11) Initialize the pins with TIOR.
(12) Restart operation by setting TSTR.

(27) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 22.152 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

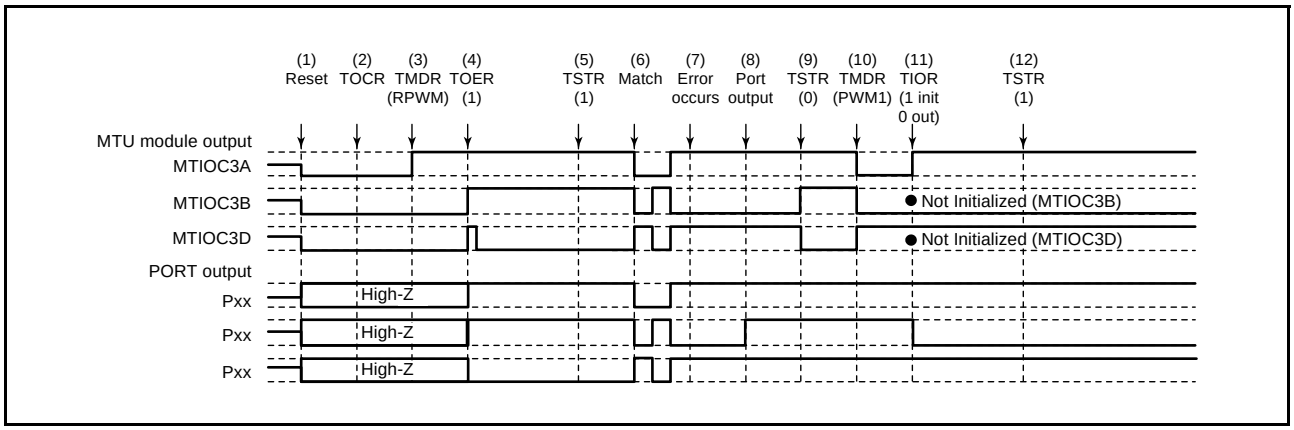


Figure 22.152 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 22.151.
(10) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).
(11) Initialize the pins with TIOR. (In PWM mode 1, the MTIOC3B side is not initialized.)
(12) Restart operation by setting TSTR.

(28) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.153 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

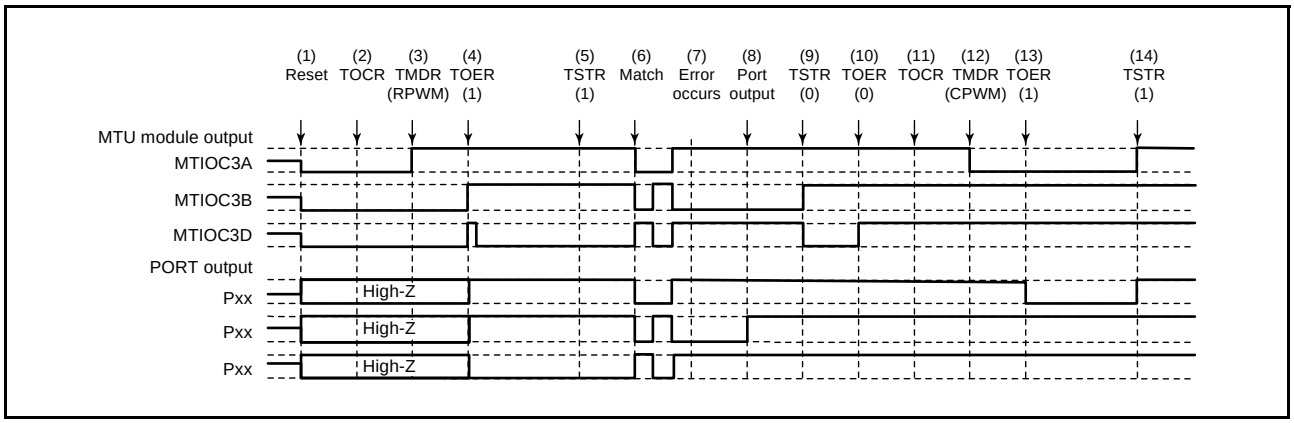


Figure 22.153 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 22.151.
(10) Disable output in MTU3 and MTU4 with TOER.
(11) Select the complementary PWM output level and enable or disable cyclic output with TOCR.
(12) Set complementary PWM mode (MTU cyclic output pin goes low).
(13) Enable output in MTU3 and MTU4 with TOER.
(14) Restart operation by setting TSTR.

(29) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.154 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.

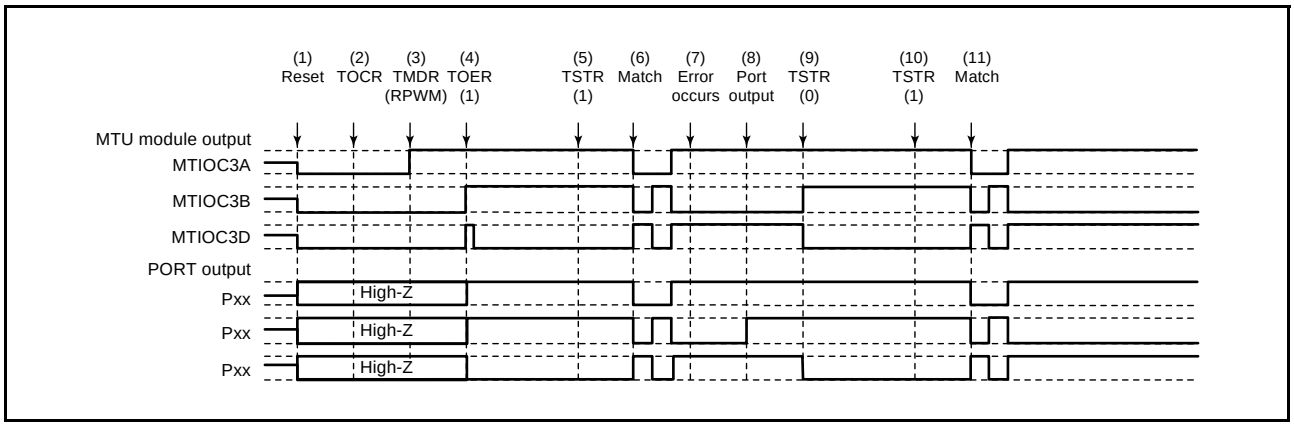


Figure 22.154 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (9) are the same as in Figure 22.151.
(10) Restart operation by setting TSTR.
(11) The reset-synchronized PWM waveform is output on compare match occurrence.

23. Port Output Enable 2 (POE2a)

The port output enable 2 (POE) module can be used to place the states of the pins for complementary PWM output by the MTU (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D), and the states of pins for MTU0 (MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D) in the high-impedance in response to changes in the input levels on the POE0# to POE3# and POE8# pins, in the output levels on pins for complementary PWM output by the MTU, oscillation stop detection by the clock generation circuit, and changes to register settings (SPOER).
It can also generate simultaneous interrupt requests.

23.1 Overview

Table 23.1 lists the specifications of the POE, and Figure 23.1 shows a block diagram of the POE.

Table 23.1 POE Specifications

Item	Description
High-impedance is controlled by the input level detection	• Falling-edge detection or sampling of the low level 16 times at PCLK/8, PCLK/16, or PCLK/128 can be set for each of the POE0# to POE3# and POE8# input pins. • Pins for complementary PWM output from the MTU can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE0# to POE3# pins. • Pins for output from MTU0 can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE8# pin.
High-impedance is controlled by the output level comparison	• Levels output on pins for complementary PWM output from the MTU are compared, and when simultaneous output of the active level continues for one or more cycles, the pins can be placed in the high-impedance.
High-impedance is controlled by the oscillation stop detection	• Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance when oscillation by the clock generation circuit stops.
High-impedance is controlled by software (registers)	• Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance by modifying settings of POE registers.
Interrupts	• Interrupts are generated in response to the results of POE0# to POE3# and POE8# input-level detection and MTU complementary PWM output-level comparison. • Interrupts are generated when pins for complementary PWM output from the MTU and output pins for MTU0 are placed in the high-impedance due to oscillation stop by the clock generation circuit.

The POE has input-level detection circuits, output-level comparison circuits, an input for the oscillation-stopped detection signal from the clock generation circuit, and a high-impedance request/ interrupt request generating circuit as shown in Figure 23.1.

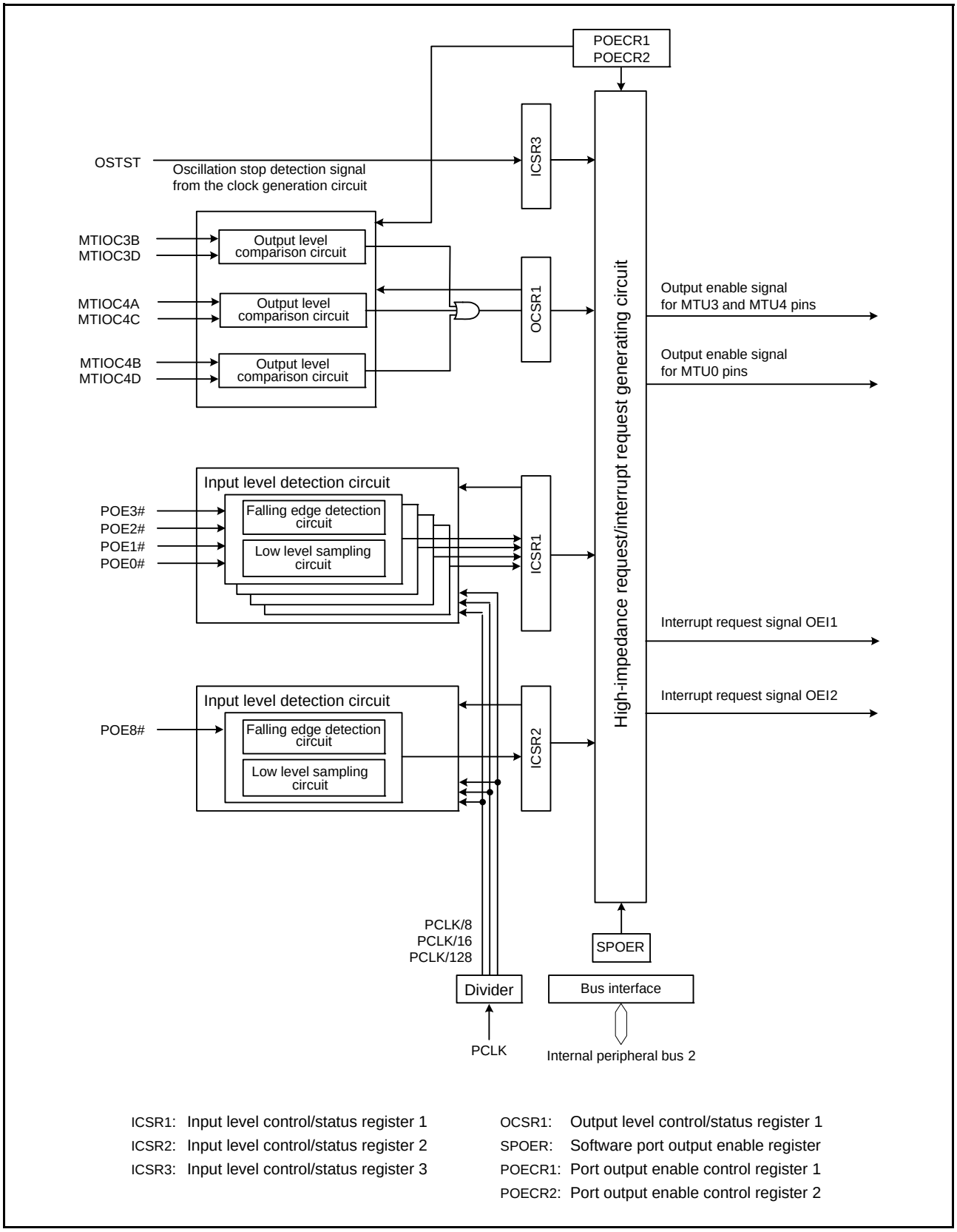


Figure 23.1 POE Block Diagram

Table 23.2 lists input/output pins to be used by the POE.

Table 23.2 LSI Input/Output Pins to be Used by POE

Pin Name	I/O	Description
POE0# to POE3#	Input	Request signals to place the pins for MTU complementary PWM output in high-impedance.
POE8#	Input	Request signals to place the MTU0 pins in high-impedance.
MTIOC3B	Output	MTU3 complementary PWM output pin
MTIOC3D	Output	MTU3 complementary PWM output pin
MTIOC4A	Output	MTU4 complementary PWM output pin
MTIOC4B	Output	MTU4 complementary PWM output pin
MTIOC4C	Output	MTU4 complementary PWM output pin
MTIOC4D	Output	MTU4 complementary PWM output pin
MTIOC0A	Output	MTU0 output pin
MTIOC0B	Output	MTU0 output pin
MTIOC0C	Output	MTU0 output pin
MTIOC0D	Output	MTU0 output pin

Table 23.3 lists output-level comparisons with pin combinations.

Table 23.3 Pin Combinations

Pin Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	Pin combinations for output-level comparison and high-impedance control can be selected by POE registers. The pins for MTU complementary PWM output are placed in high-impedance when the pins simultaneously output an active level for one or more cycles of PCLK. (When the MTU. TOCR1.TOCS bit = 0: Low level if the MTU.TOCR1.OLSP and OSLN bits are 0, and high level if the MTU.TOCR1.OLSP and OSLN bits are 1. When the MTU. TOCR1.TOCS bit = 1: Low level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N and OLS1P bits are 0, and high level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N and OLS1P bits are 1.)
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	

23.2 Register Descriptions

23.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): 0008 8900h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE1	POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		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# input. 0 1: Accepts a request when POE0# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE0# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE0# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b3, b2	POE1M[1:0]	POE1 Mode Select	b3 b2 0 0: Accepts a request on the falling edge of POE1# input. 0 1: Accepts a request when POE1# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE1# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE1# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b5, b4	POE2M[1:0]	POE2 Mode Select	b5 b4 0 0: Accepts a request on the falling edge of POE2# input. 0 1: Accepts a request when POE2# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE2# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE2# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7, b6	POE3M[1:0]	POE3 Mode Select	b7 b6 0 0: Accepts a request on the falling edge of POE3# input. 0 1: Accepts a request when POE3# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE3# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE3# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b8	PIE1	Port Interrupt Enable 1	0: OEI1 interrupt requests by the input level detection disabled 1: OEI1 interrupt requests by the input level detection 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
b13	POE1F	POE1 Flag	0: Indicates that a high impedance request has not been input to the POE1# pin. 1: Indicates that a high impedance request has been input to the POE1# pin.	R/(W)*2
b14	POE2F	POE2 Flag	0: Indicates that a high impedance request has not been input to the POE2# pin. 1: Indicates that a high impedance request has been input to the POE2# pin.	R/(W)*2
b15	POE3F	POE3 Flag	0: Indicates that a high impedance request has not been input to the POE3# pin. 1: Indicates that a high impedance request has been input to the POE3# pin.	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.

When low-level sampling has been set by the POE0M[1:0] to POE3M[1:0] bits, writing 0 to the POE0F to POE3F flags requires high level input on the POE0# to POE3# pins. For details, see section 23.3.5, Release from the High-Impedance.

PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables OEI1 interrupt requests when any one of the POE0F to POE3F flags is set to 1.

POE0F Flag (POE0 Flag)

This flag indicates that a high impedance request has been input to the POE0# pin.
[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1

[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

POE1F Flag (POE1 Flag)

This flag indicates that a high impedance request has been input to the POE1# pin.
[Clearing condition]

- By writing 0 to POE1F after reading POE1F = 1

[Setting condition]

- When the input set by POE1M[1:0] occurs at the POE1# pin

POE2F Flag (POE2 Flag)

This flag indicates that a high impedance request has been input to the POE2# pin.
[Clearing condition]

- By writing 0 to POE2F after reading POE2F = 1

[Setting condition]

- When the input set by POE2M[1:0] occurs at the POE2# pin

POE3F Flag (POE3 Flag)

This flag indicates that a high impedance request has been input to the POE3# pin.
[Clearing condition]

- By writing 0 to POE3F after reading POE3F = 1

[Setting condition]

- When the input set by POE3M[1:0] occurs at the POE3# pin

23.2.2 Output Level Control/Status Register 1 (OCSR1)

Address(es): 0008 8902h

	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: OEI1 interrupt requests by the output level comparison disabled 1: OEI1 interrupt requests by the output level comparison enabled	R/W
b9	OCE1	Output Short High-Impedance Enable 1	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	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.

OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables OEI1 interrupt requests when the OSF1 flag is set to 1.

OCE1 Bit (Output Short High-Impedance Enable 1)

This bit specifies whether to place the MTU complementary PWM output pins in high-impedance 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 outputs for MTU complementary PWM output to be compared in Table 23.3 has simultaneously become an active level. If the PnCZEA (n = 1, 2, 3) bits in POECR2 are 0 or the output comparison function of the MTU is not enabled, the OSF1 flag will not be set to 1 even if both pins in the corresponding complementary output pair of the MTU are simultaneously active.

[Clearing conditions]

- By writing 0 to OSF1 after reading OSF1 = 1

The complementary output pins for the MTU must be at the inactive level when 0 is written to the flag. For details, see section 23.3.5, Release from the High-Impedance.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

23.2.3 Input Level Control/Status Register 2 (ICSR2)

Address(es): 0008 8908h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE8F	—	—	POE8E	PIE2	—	—	—	—	—	—	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# input 0 1: Accepts a request when POE8# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE8# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE8# 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: OEI2 interrupt requests disabled 1: OEI2 interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance. 1: Places the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance.	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.

PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables OEI2 interrupt requests when the POE8F flag is set to 1.

POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to place the MTU0 pins in high-impedance 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.
[Clearing condition]
Writing 0 to POE8F after reading POE8F = 1
When writing 0 to the flag while low-level sampling is selected for the POE8M[1:0] bits, the POE8# pin input must be at the high level.
For details, see section 23.3.5, Release from the High-Impedance.
[Setting condition]

- When the input set by POE8M[1:0] occurs at the POE8# pin

23.2.4 Software Port Output Enable Register (SPOER)

Address(es): 0008 890Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CH0HIZ	CH34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CH34HIZ	MTU3 and MTU4 Output High-Impedance Enable	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W
b1	CH0HIZ	MTU0 Output High-Impedance Enable	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CH34HIZ Bit (MTU3 and MTU4 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance.

[Clearing condition]

- By writing 0 to CH34HIZ after reading CH34HIZ = 1

[Setting condition]

- By writing 1 to CH34HIZ

CH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit specifies whether to place the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) in high-impedance.

[Clearing condition]

- By writing 0 to CH0HIZ after reading CH0HIZ = 1

[Setting condition]

- By writing 1 to CH0HIZ

23.2.5 Port Output Enable Control Register 1 (POECR1)

Address(es): 0008 890Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PE3ZE	PE2ZE	PE1ZE	PE0ZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PE0ZE	MTIOC0A High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b1	PE1ZE	MTIOC0B High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b2	PE2ZE	MTIOC0C High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b3	PE3ZE	MTIOC0D High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	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.

23.2.6 Port Output Enable Control Register 2 (POECR2)

Address(es): 0008 890Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	P1CZEA	P2CZEA	P3CZEA	—	—	—	—
Value after reset:	0	1	1	1	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	P3CZEA	MTU Port 3 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b5	P2CZEA	MTU Port 2 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b6	P1CZEA	MTU Port 1 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

When this function is not used, write 00h to this register.

P3CZEA Bit (MTU Port 3 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4B and MTIOC4D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4B and MTIOC4D pins are compared.

P2CZEA Bit (MTU Port 2 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4A and MTIOC4C pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4A and MTIOC4C pins are compared.

P1CZEA Bit (MTU Port 1 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC3B and MTIOC3D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC3B and MTIOC3D pins are compared.

23.2.7 Input Level Control/Status Register 3 (ICSR3)

Address(es): 0008 890Eh

	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 place the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance. 1: Places the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance.	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: Oscillation stop is not producing a request to place pins in the high-impedance. 1: Oscillation stop is producing a request to place pins in the high-impedance.	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.

OSTSTE Bit (OSTST High-Impedance Enable)

This bit permits or prohibits placement of pins for complementary PWM output from MTU and output pins for MTU0 in the high-impedance on detection that oscillation has stopped.

OSTSTF Flag (OSTST High-Impedance Flag)

The OSTSTF flag is a status flag that indicates the state of requests to place pins in the high-impedance due to oscillation having stopped. The value of the flag becomes 1 when oscillation stops. Ensure that the oscillation-stopped detection signal is negated when clearing the flag by writing 0 to it. Writing 0 to the OSTSTF flag while the oscillation-stopped detection signal is being asserted will not clear the flag. The assertion of the oscillation-stopped detection signal takes 10 cycles of PCLK after detecting the oscillation stop.

[Clearing condition]

- Writing 0 to the bit after having read its value as 1.

[Setting condition]

- Detection of the oscillation-stopped state

23.3 Operation

The target pins for high-impedance control and conditions to place the pins in high-impedance are described below.

(1) MTU0 pin (MTIOC0A)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When ICSR2.POE8F flag is set to 1 with POECR1.PE0ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE0ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE0ZE and ICSR3.OSTSTE set to 1.

(2) MTU0 pin (MTIOC0B)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When ICSR2.POE8F flag is set to 1 with POECR1.PE1ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE1ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE1ZE and ICSR3.OSTSTE set to 1.

(3) MTU0 pin (MTIOC0C)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When ICSR2.POE8F flag is set to 1 with POECR1.PE2ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE2ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE2ZE and ICSR3.OSTSTE set to 1.

(4) MTU0 pin (MTIOC0D)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When ICSR2.POE8F flag is set to 1 with POECR1.PE3ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE3ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE3ZE and ICSR3.OSTSTE set to 1.

(5) MTU3 pins (MTIOC3B and MTIOC3D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection
  When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P1CZEA set to 1.
- MTIOC3B and MTIOC3D output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P1CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P1CZEA set to 1.

- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P1CZEA and ICSR3.OSTSTE set to 1.

(6) MTU4 pins (MTIOC4A and MTIOC4C)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection
  When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P2CZEA set to 1.
- MTIOC4A and MTIOC4C output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P2CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P2CZEA set to 1.
- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P2CZEA and ICSR3.OSTSTE set to 1.

(7) MTU4 pins (MTIOC4B and MTIOC4D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

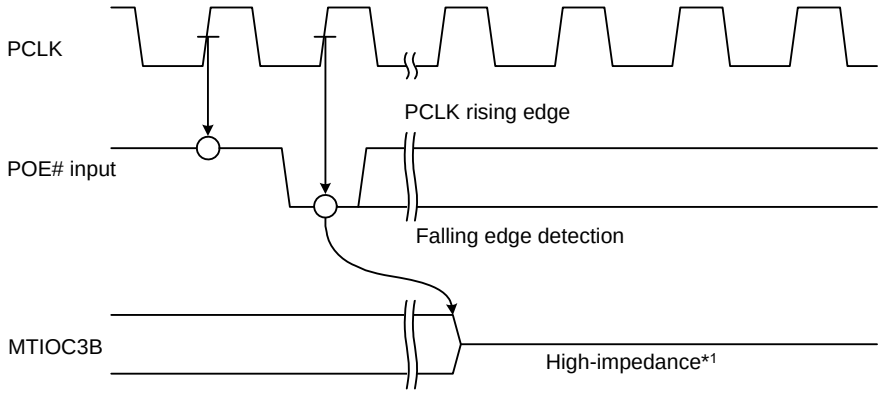
- POE0# to POE3# input level detection
  When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P3CZEA set to 1.
- MTIOC4B and MTIOC4D output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P3CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P3CZEA set to 1.
- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P3CZEA and ICSR3.OSTSTE set to 1.

23.3.1 Input Level Detection Operation

If the input conditions set by ICSR1 and ICSR2 occur on the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.
A falling edge is detected after PCLK causes sampling to proceed. If the low level is input to the POE0# to POE3# or POE8# pin over less than one full cycle of PCLK, whether the falling edge will or will not be detected cannot be guaranteed.
Figure 23.2 shows the timing of sampling after the level changes in input to the POE0# to POE3# and POE8# pins until the respective pins enter high-impedance.

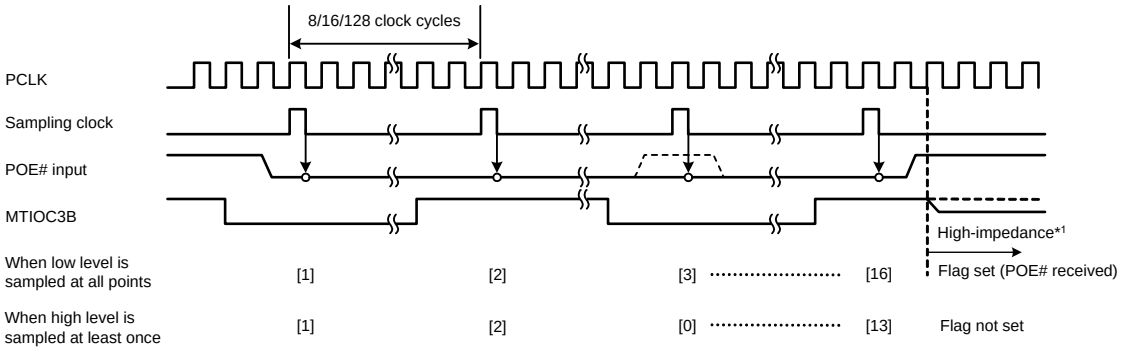


Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

Figure 23.2 Falling Edge Detection

(2) Low-Level Detection

Figure 23.3 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock selected by ICSR1 and ICSR2. If even one high level is detected during this interval, the low level is not accepted. Furthermore, in an interval over which the sampling clock is not being output, changes to the levels on the POE0# to POE3# and POE8# pins are ignored.



Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

Figure 23.3 Low-Level Detection Operation

23.3.2 Output-Level Compare Operation

Figure 23.4 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D (MTU complementary PWM output pins). The operation is the same for the other pin combinations.

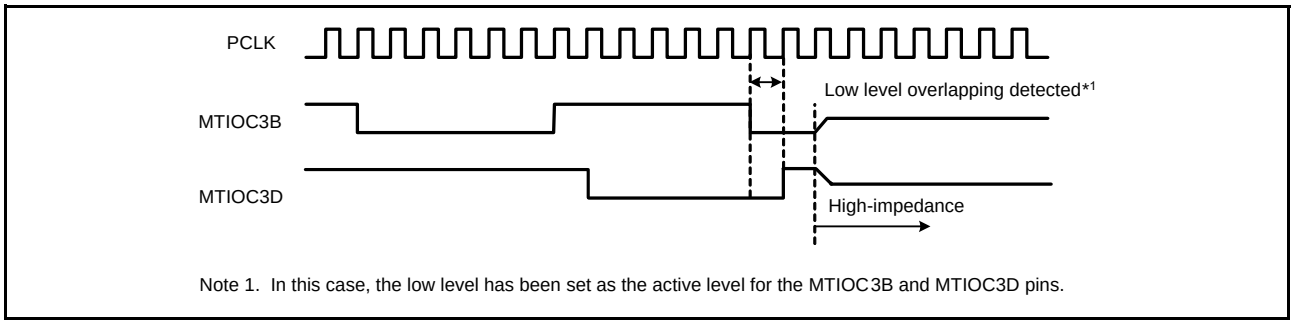


Figure 23.4 Output-Level Compare Operation

23.3.3 High-Impedance Control Using Registers

The high-impedance of the MTU complementary PWM output and MTU0 pins can be directly controlled using the software port output enable register (SPOER).
Setting the CH34HIZ bit in SPOER to 1 places the MTU complementary PWM output pins (MTU3 and MTU4) specified by the port output enable control register 2 (POECR2) in the high-impedance.
The high-impedance of other pins can also be controlled by setting the appropriate bits in SPOER.

23.3.4 High-Impedance Control on Detection of Stopped Oscillation

When the oscillation-stop detection circuit in the clock pulse generator detects stopped oscillation, pins for complementary PWM output from MTU and pins for MTU0 as specified by the input level control and status register 3 (ICSR3) can be placed in the high-impedance.

23.3.5 Release from the High-Impedance

Pins for complementary PWM output from MTU and pins for MTU0 which have been placed in the high-impedance due to input-level detection can be released from that state by either returning them to their initial state with a reset or clearing all of the POE3F to POE0F flags in ICSR1 and the POE8F flag in ICSR2. Note, however, that when low-level sampling is selected by the POE3M[1:0] to POE0M[1:0] bits in ICSR1 and the POE8M[1:0] bits in ICSR2, if a high level is being input to the corresponding pin from among POE0# to POE3# and POE#8 but has not yet been sampled, writing 0 to the flag is ignored (the flag is not cleared).
MTU complementary PWM output pins which have been placed in the high-impedance due to output-level comparison can be released from that state by either returning them to their initial state with a reset or clearing the OSF1 flag in OCSR1. Note, however, that if the inactive level is not yet being output from the MTU complementary PWM output pins, writing 0 to the flag is ignored (the flag is not cleared). Inactive-level outputs can be obtained by setting the MTU registers.
For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance because oscillation by the clock generation circuit has stopped, clearing the OSTSTF or OSTSTE bit in ICSR3 releases the pins from the high-impedance.
For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance by the SPOER.CH34HIZ or SPOER.CH0HIZ bit, clearing the corresponding bits (SPOER.CH34HIZ and SPOER.CH0HIZ) releases the pins from the high-impedance.

23.4 Interrupts

The POE issues a request to generate an interrupt when the corresponding condition below is matched during input-level detection, output-level comparison, or oscillation stop by the clock generation circuit. Table 23.4 shows the interrupt sources and their request conditions.
On acceptance of an OEI1 or OEI2 interrupt, the first line of the exception handling routine for the given interrupt should confirm that the flag for the given flag has been set to 1.

Table 23.4 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F, POE1F, POE2F, POE3F, OSF1	When ICSR1.POE0F, POE1F, POE2F, or POE3F flag is set to 1 with ICSR1.PIE1 set to 1, or when OCSR1.OSF1 flag is set to 1 with OCSR1.OIE1 set to 1.
OEI2	Output enable interrupt 2	POE8F	When ICSR2.POE8F flag is set to 1 with ICSR2.PIE2 set to 1.

23.5 Usage Notes

23.5.1 Transitions to Software Standby Mode or Deep Standby 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 of pins cannot be controlled.

23.5.2 When POE is not Used

When the POE is not used, write 00h to the port output enable control registers 1 and 2 (POECR1 and POECR2), respectively.

23.5.3 Specifying Pins corresponding to the MTU

The POE controls high-impedance outputs only when a pin has been specified so that the pin corresponds to the MTU by setting the port mode register (PMR). When the pin has been specified as a general I/O pin, the POE does not control high-impedance outputs.

24. 16-Bit Timer Pulse Unit (TPUa)

The RX63N/RX631 Group has two on-chip 16-bit timer pulse units (TPU), unit 0 and unit 1, each comprising six channels. Therefore, this LSI includes twelve channels (TPU0 to TPU11).

24.1 Overview

Specifications of the TPU are listed in Table 24.1. Functions of TPU (unit 0) and TPU (unit 1) are listed in Table 24.2 and Table 24.3, respectively.
Block diagrams of TPU (unit 0) and TPU (unit 1) are shown in Figure 24.1 and Figure 24.2, respectively.

Table 24.1 Specifications of TPU

Item	Description
Pulse input/output	Maximum 32 (for unit 0: 16, for unit 1: 16)
Count clock	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
Channels 0, 3, 6, and 9	Buffer operation can be set.
Channels 1, 2, 4, 5, 7, 8, 10, and 11	Phase counting mode can be set
Interrupt source	52 sources (for unit 0: 26, for unit 1: 26)
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.
Low-power consumption function	Module stop state can be set for each unit.

Table 24.2 TPU (Unit 0) Functions (1/2)

Item		TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
Count clock		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
Timer general registers (TGRy) (y = A to D)		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		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

Table 24.2 TPU (Unit 0) Functions (2/2)

Item	TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
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	—	Possible	Possible
Buffer operation	Possible	Not possible	Not possible	Possible	Not possible	Not possible
DTC activation	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 (TGRy) (y = A to D)	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
	TGRy compare match or input capture	Not possible	Not possible	Not possible	Not possible	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
Module stop setting*2	MSTPA13 bit in MSTPCRA					

Note 1. TGRC and TGRD can be set as a buffer register.
Note 2. For details, see section 11, Low Power Consumption.

Table 24.3 TPU (Unit 1) Functions

Item		TPU6	TPU7	TPU8	TPU9	TPU10	TPU11
Count clock		PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKE TCLKF TCLKG TCLKH	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKE TCLKF	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKE TCLKF TCLKG	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKE	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKE TCLKG	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKE TCLKG TCLKH
Timer general registers (TGRy) (y = A to D)		TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB
I/O pins		TIOCA6 TIOCB6 TIOCC6 TIOCD6	TIOCA7 TIOCB7	TIOCA8 TIOCB8	TIOCA9 TIOCB9 TIOCC9 TIOCD9	TIOCA10 TIOCB10	TIOCA11 TIOCB11
Counter clear function		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		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 (TGRy) (y = A to D)		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
		TGRy compare match or input capture	Not possible	Not possible	Not possible	Not possible	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 6A • Compare match or input capture 6B • Compare match or input capture 6C • Compare match or input capture 6D • Overflow	4 sources • Compare match or input capture 7A • Compare match or input capture 7B • Overflow • Underflow	4 sources • Compare match or input capture 8A • Compare match or input capture 8B • Overflow • Underflow	5 sources • Compare match or input capture 9A • Compare match or input capture 9B • Compare match or input capture 9C • Compare match or input capture 9D • Overflow	4 sources • Compare match or input capture 10A • Compare match or input capture 10B • Overflow • Underflow	4 sources • Compare match or input capture 11A • Compare match or input capture 11B • Overflow • Underflow
Module stop setting*2		MSTPA12 bit in MSTPCRA					

Note 1. TGRC and TGRD can be set as a buffer register.
Note 2. For details, see section 11, Low Power Consumption.

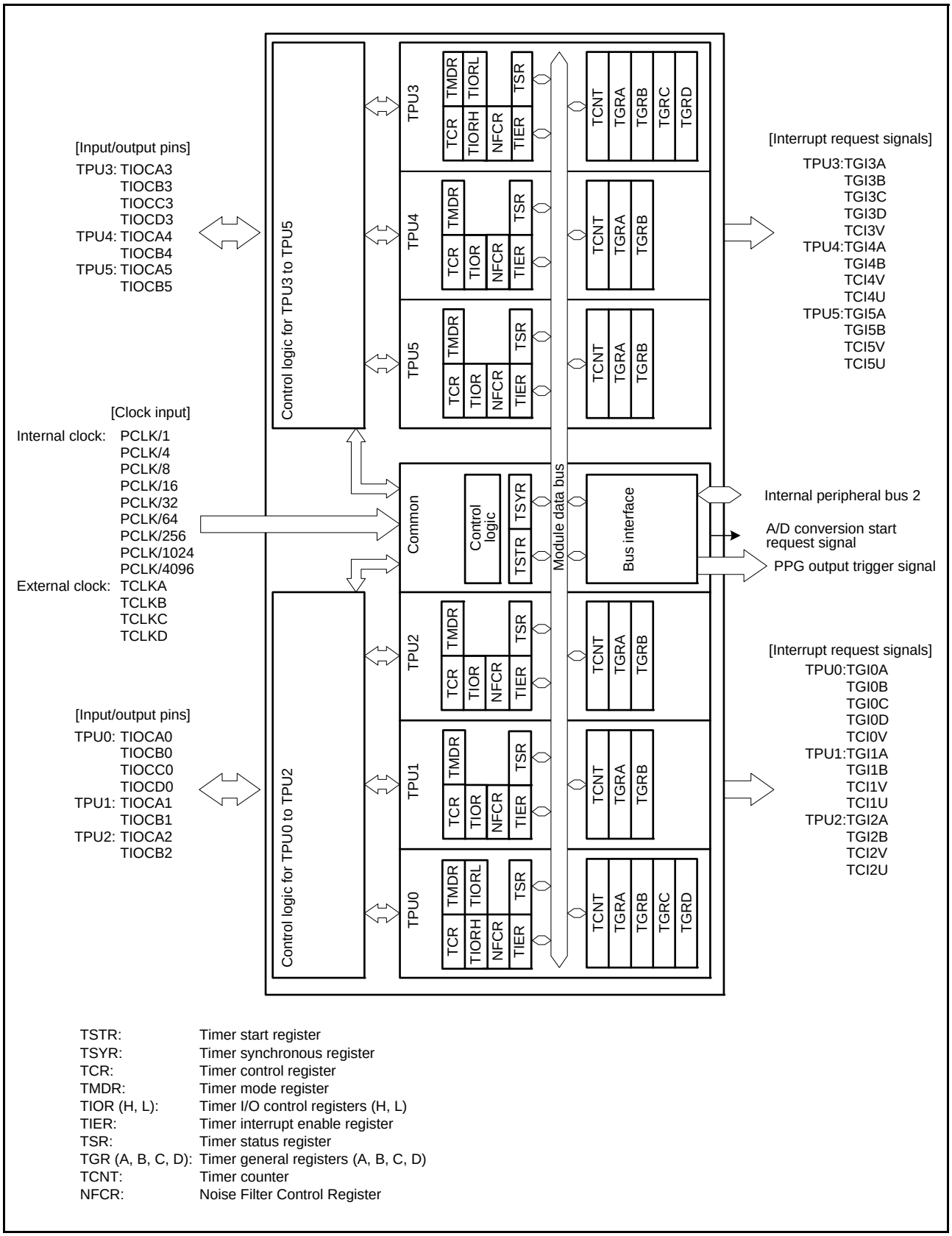


Figure 24.1 Block Diagram of TPU (Unit 0)

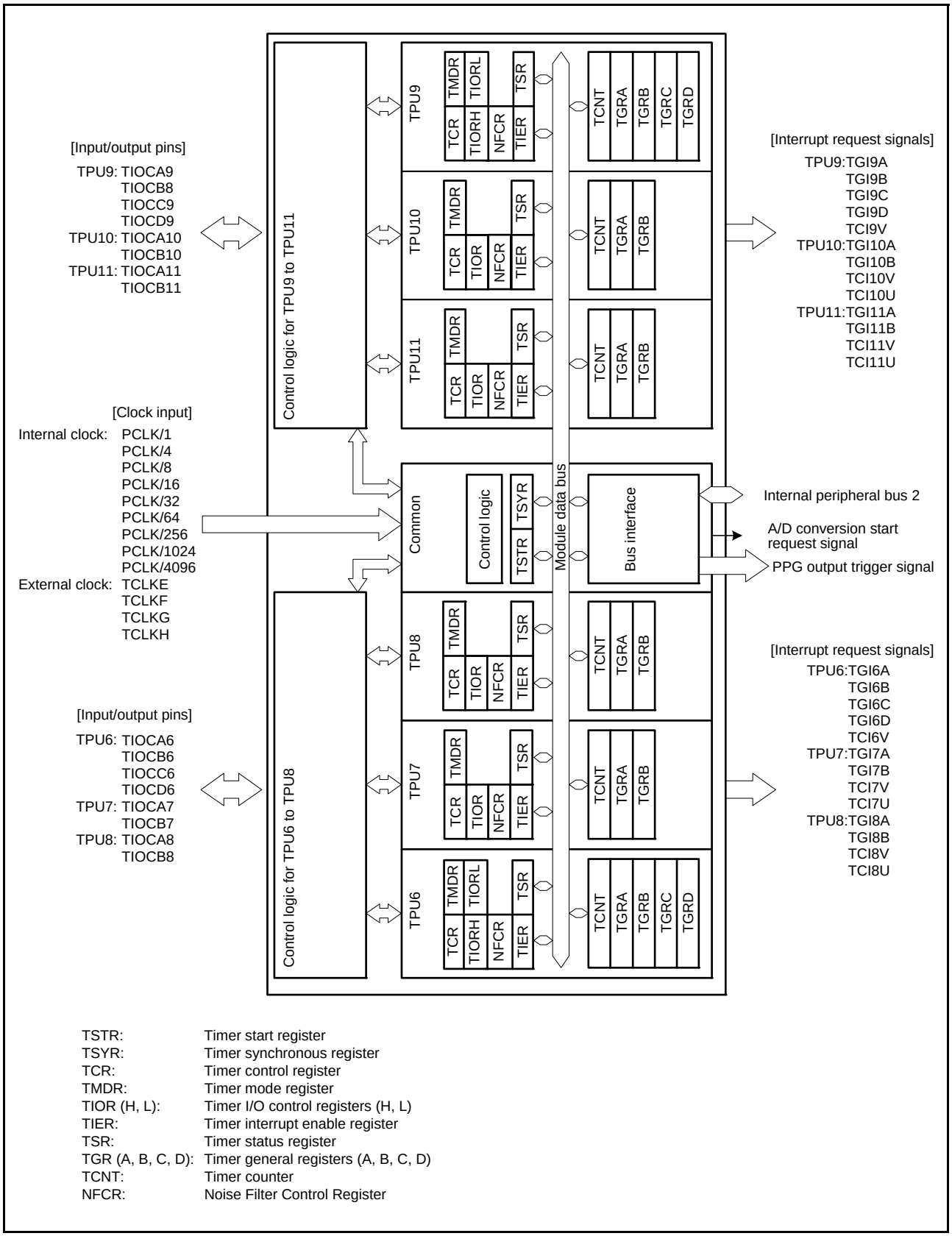


Figure 24.2 Block Diagram of TPU (Unit 1)

Table 24.4 lists the input/output pins of the TPU.

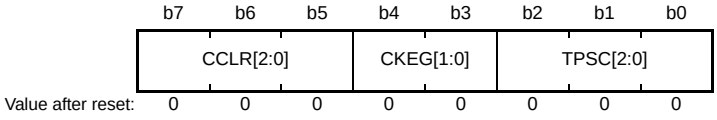
Table 24.4 Pin Configuration of TPU

Unit	Channel	Pin Name	I/O	Description
Unit 0	Common to unit 0	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
Unit 1	Common to unit 1	TCLKE	Input	External clock E input pin (TPU7 and TPU11 phase counting mode A phase input)
		TCLKF	Input	External clock F input pin (TPU7 and TPU11 phase counting mode B phase input)
		TCLKG	Input	External clock G input pin (TPU8 and TPU10 phase counting mode A phase input)
		TCLKH	Input	External clock H input pin (TPU8 and TPU10 phase counting mode B phase input)
	TPU6	TIOCA6	I/O	TPU6.TGRA input capture input/output compare output/PWM output pin
		TIOCB6	I/O	TPU6.TGRB input capture input/output compare output/PWM output pin
		TIOCC6	I/O	TPU6.TGRC input capture input/output compare output/PWM output pin
		TIOCD6	I/O	TPU6.TGRD input capture input/output compare output/PWM output pin
	TPU7	TIOCA7	I/O	TPU7.TGRA input capture input/output compare output/PWM output pin
		TIOCB7	I/O	TPU7.TGRB input capture input/output compare output/PWM output pin
	TPU8	TIOCA8	I/O	TPU8.TGRA input capture input/output compare output/PWM output pin
		TIOCB8	I/O	TPU8.TGRB input capture input/output compare output/PWM output pin
	TPU9	TIOCA9	I/O	TPU9.TGRA input capture input/output compare output/PWM output pin
		TIOCB9	I/O	TPU9.TGRB input capture input/output compare output/PWM output pin
		TIOCC9	I/O	TPU9.TGRC input capture input/output compare output/PWM output pin
		TIOCD9	I/O	TPU9.TGRD input capture input/output compare output/PWM output pin
	TPU10	TIOCA10	I/O	TPU10.TGRA input capture input/output compare output/PWM output pin
		TIOCB10	I/O	TPU10.TGRB input capture input/output compare output/PWM output pin
	TPU11	TIOCA11	I/O	TPU11.TGRA input capture input/output compare output/PWM output pin
		TIOCB11	I/O	TPU11.TGRB input capture input/output compare output/PWM output pin

24.2 Register Descriptions

24.2.1 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, TPU6.TCR 0008 8180h, TPU7.TCR 0008 8190h, TPU8.TCR 0008 81A0h, TPU9.TCR 0008 81B0h, TPU10.TCR 0008 81C0h, TPU11.TCR 0008 81D0h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Timer Prescaler Select	See Table 24.5 to Table 24.10.	R/W
b4, b3	CKEG[1:0]	Input Clock Edge Select	See Table 24.11.	R/W
b7 to b5	CCLR[2:0] *1	Counter Clear Source Select	See Table 24.12 and Table 24.13.	R/W

Note 1. Bit 7 in TCR of TPU1, TPU2, TPU4, and TPU5 of unit 0 and bit 7 in TCR of TPU7, TPU8, TPU10, and TPU11 of unit 1 are reserved. These bits are read as 0. The write value should be 0.

TPUm.TCR settings should be made while TPUm.TCNT counter operation is stopped.

TPSC[2:0] Bits (Timer Prescaler Select)

These bits select the TCNT counter 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 20, 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 24.5 Bits TPSC[2:0] (TPU0, TPU6)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU0 (unit 0) TPU6 (unit 1)	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 • TPU0 (unit 0): counts on TCLKA pin input • TPU6 (unit 1): counts on TCLKE pin input
	1	0	1	External clock • TPU0 (unit 0): counts on TCLKB pin input • TPU6 (unit 1): counts on TCLKF pin input
	1	1	0	External clock • TPU0 (unit 0): counts on TCLKC pin input • TPU6 (unit 1): counts on TCLKG pin input
	1	1	1	External clock • TPU0 (unit 0): counts on TCLKD pin input • TPU6 (unit 1): counts on TCLKH pin input

Table 24.6 Bits TPSC[2:0] (TPU1, TPU7)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU1 (unit 0) TPU7 (unit 1)	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 • TPU1 (unit 0): counts on TCLKA pin input • TPU7 (unit 1): counts on TCLKE pin input
	1	0	1	External clock • TPU1 (unit 0): counts on TCLKB pin input • TPU7 (unit 1): counts on TCLKF pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	• TPU1 (unit 0) Counts on TPU2.TCNT counter overflow/underflow • TPU7 (unit 1) Counts on TPU8.TCNT counter overflow/underflow

Note 1. This setting is invalid when TPU1 or TPU7 is in phase counting mode.

Table 24.7 Bits TPSC[2:0] (TPU2, TPU8)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU2 (unit 0) TPU8 (unit 1)	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 • TPU2 (unit 0): counts on TCLKA pin input • TPU8 (unit 1): counts on TCLKE pin input
	1	0	1	External clock • TPU2 (unit 0): counts on TCLKB pin input • TPU8 (unit 1): counts on TCLKF pin input
	1	1	0	External clock • TPU2 (unit 0): counts on TCLKC pin input • TPU8 (unit 1): counts on TCLKG pin input
	1	1	1	Internal clock: counts on PCLK/1024

Note 1. This setting is invalid when TPU2 or TPU8 is in phase counting mode.

Table 24.8 Bits TPSC[2:0] (TPU3, TPU9)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU3 (unit 0) TPU9 (unit 1)	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 • TPU3 (unit 0): counts on TCLKA pin input • TPU9 (unit 1): counts on TCLKE 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 24.9 Bits TPSC[2:0] (TPU4, TPU10)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU4 (unit 0) TPU10 (unit 1)	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 • TPU4 (unit 0): counts on TCLKA pin input • TPU10 (unit 1): counts on TCLKE pin input
	1	0	1	External clock • TPU4 (unit 0): counts on TCLKC pin input • TPU10 (unit 1): counts on TCLKG pin input
	1	1	0	Internal clock: counts on PCLK/1024
	1	1	1	• TPU4 (unit 0) Counts on TPU5.TCNT counter overflow/underflow • TPU10 (unit 1) Counts on TPU11.TCNT counter overflow/underflow

Note 1. This setting is invalid when TPU4 or TPU10 is in phase counting mode.

Table 24.10 Bits TPSC[2:0] (TPU5, TPU11)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU5 (unit 0) TPU11 (unit 1)	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 • TPU5 (unit 0): counts on TCLKA pin input • TPU11 (unit 1): counts on TCLKE pin input
	1	0	1	External clock • TPU5 (unit 0): counts on TCLKC pin input • TPU11 (unit 1): counts on TCLKG pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	External clock • TPU5 (unit 0): counts on TCLKD pin input • TPU11 (unit 1): counts on TCLKH pin input

Note 1. This setting is invalid when TPU5 or TPU11 is in phase counting mode.

Table 24.11 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 24.12 Bits CCLR[2:0] (TPU0, TPU3, TPU6, TPU9)

Channel	Bits CCLR[2:0]			Description
	b7	b6	b5	
(Unit 0) TPU0, TPU3 (Unit 1) TPU6, TPU9	0	0	0	TCNT counter clearing disabled
	0	0	1	TCNT counter cleared by TGRA compare match/input capture
	0	1	0	TCNT counter cleared by TGRB compare match/input capture
	0	1	1	TCNT counter cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	TCNT counter clearing disabled
	1	0	1	TCNT counter cleared by TGRC compare match/input capture*1
	1	1	0	TCNT counter cleared by TGRD compare match/input capture*1
	1	1	1	TCNT counter 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 counter 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) and the TPUB.TSYR.SYNCj bit (j = 6, 9) to 1.

Table 24.13 Bits CCLR[2:0] (TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, TPU11)

Channel	Bits CCLR[2:0]*1			Description
	b7	b6	b5	
(Unit 0) TPU1, TPU2, TPU4, TPU5 (Unit 1) TPU7, TPU8, TPU10, TPU11	0	0	0	TCNT counter clearing disabled
	0	0	1	TCNT counter cleared by TGRA compare match/input capture
	0	1	0	TCNT counter cleared by TGRB compare match/input capture
	0	1	1	TCNT counter 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 of unit 0 and bit 7 in TCR of TPU7, TPU8, TPU10, and TPU11 of unit 1 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) and the TPUB.TSYR.SYNCj bit (j = 7, 8, 10, 11) to 1.

24.2.2 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, TPU6.TMDR 0008 8181h, TPU7.TMDR 0008 8191h, TPU8.TMDR 0008 81A1h, TPU9.TMDR 0008 81B1h, TPU10.TMDR 0008 81C1h, TPU11.TMDR 0008 81D1h

	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*1 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*2 0 1 0 1: Phase counting mode 2*2 0 1 1 0: Phase counting mode 3*2 0 1 1 1: Phase counting mode 4*2 Settings other than above are prohibited.	R/W
b4	BFA*3	Buffer Operation A	0: TPUm.TGRA operates normally 1: TPUm.TGRA and TPUm.TGRC used together for buffer operation (m = 0, 3, 6, 9)	R/W
b5	BFB*4	Buffer Operation B	0: TPUm.TGRB operates normally 1: TPUm.TGRB and TPUm.TGRD used together for buffer operation (m = 0, 3, 6, 9)	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 11)	R/W
b7	ICSELD*4	GRD Input Capture Input Select	0: Input capture input source is TIOCDn pin 1: Input capture input source is TIOCCn pin (n = 0, 3, 6, 9)	R/W

Note 1. Bit 3 is reserved. This bit is read as 0. The write value should be 0.
Note 2. Phase counting mode cannot be set for TPU0, TPU3 (unit 0), TPU6, and TPU9 (unit 1). A 0 should be written to bit 2 for them.
Note 3. Bit 4 of TPU1, TPU2, TPU4, TPU5 (unit 0), TPU7, TPU8, TPU10, and TPU11 (unit 1) that do not have TGRC is reserved. This bit is read as 0. The write value should be 0.
Note 4. Bits 5 and 7 of TPU1, TPU2, TPU4, TPU5 (unit 0), TPU7, TPU8, TPU10, and TPU11 (unit 1) 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 counter operation is stopped.

BFA Bit (Buffer Operation A)

Specifies whether TPUm.TGRA (m = 0, 3, 6, 9) is to normally operate, or TPUm.TGRA and TPUm.TGRC (m = 0, 3, 6, 9) 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, 6, 9) is to normally operate, or TPUm.TGRB and TPUm.TGRD (m = 0, 3, 6, 9) 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 11).
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)

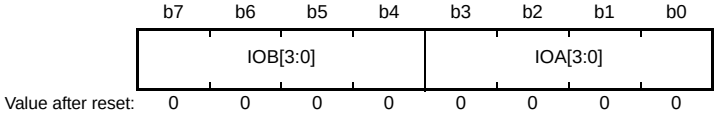
Selects the input capture input for TPUm.TGRD (m = 0, 3, 6, 9).

This function allows measurement of high-level width and period of the input pulse on a TIOCCn input pin.

24.2.3 Timer I/O Control Register (TIORH, TIORL, TIOR)

- Unit 0 (TPU0.TIORH, TPU1.TIOR, TPU2.TIOR, TPU3.TIORH, TPU4.TIOR, TPU5.TIOR)
  Unit 1 (TPU6.TIORH, TPU7.TIOR, TPU8.TIOR, TPU9.TIORH, TPU10.TIOR, TPU11.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, TPU6.TIORH 0008 8182h, TPU7.TIOR 0008 8192h, TPU8.TIOR 0008 81A2h, TPU9.TIORH 0008 81B2h, TPU10.TIOR 0008 81C2h, TPU11.TIOR 0008 81D2h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	TGRA Control	See Table 24.14 to Table 24.19.*1	R/W
b7 to b4	IOB[3:0]	TGRB Control	See Table 24.14 to Table 24.19.*1	R/W

Note 1. If the IOn[3:0] bit (n = A or 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 11) is placed in high impedance state.

- Unit 0 (TPU0.TIORL, TPU3.TIORL)
  Unit 1 (TPU6.TIORL, TPU9.TIORL)

Address(es): TPU0.TIORL 0008 8113h, TPU3.TIORL 0008 8143h, TPU6.TIORL 0008 8183h, TPU9.TIORL 0008 81B3h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	TGRC Control	See Table 24.20 and Table 24.21.*1	R/W
b7 to b4	IOD[3:0]	TGRD Control	See Table 24.20 and Table 24.21.*1	R/W

Note 1. If the IOn[3:0] bit (n = C or 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, 6, or 9) is placed in high impedance state.

TPU has four TIORH registers, one for TPU0, TPU3, TPU6, and TPU9, and four TIORL registers, one for TPU0, TPU3, TPU6, and TPU9, and also has eight TIOR registers, one for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11. Thus the TPUA has sixteen 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 24.14 to Table 24.21.
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) and the TPUB.TSTR.CSTj bit (j = 6 to 11) are cleared to 0). In PWM mode 2, the output at the time when the TCNT counter 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 corresponding set of IOC[3:0] or IOD[3:0] bits becomes 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 20, I/O Ports.

IOA[3:0] Bits (TGRA Control)

Select the function of TPUm.TGRA (m = 0 to 11).

IOB[3:0] Bits (TGRB Control)

Select the function of TPUm.TGRB (m = 0 to 11).

IOC[3:0] Bits (TGRC Control)

Select the function of TPUm.TGRC (m = 0, 3, 6, 9).

IOD[3:0] Bits (TGRD Control)

Select the function of TPUm.TGRD (m = 0, 3, 6, 9).

Table 24.14 TPU0.TIORH, TPU6.TIORH

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 0, 6) Function	TIOCAn Pin (n = 0, 6) 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 TIOCAn pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCAn pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCAn pin; input capture at both edges
1	1	x	x		• TPU0 Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*1 • TPU6 Capture input source is TPU7 count clock Input capture at TPU7.TCNT count-up/count-down*1

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 0, 6) Function	TIOCBn Pin (n = 0, 6) 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 TIOCBn or TIOCAn pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCBn or TIOCAn pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCBn or TIOCAn pin*2; input capture at both edges
1	1	x	x		• TPU0 Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*1 • TPU6 Capture input source is TPU7 count clock Input capture at TPU7.TCNT count-up/count-down*1

x: Don't care

Note 1. When the TPSC[2:0] bits in TPUm.TCR are set to 000b and PCLK/1 is used as the TPUm.TCNT count clock, this setting is invalid and input capture is not generated (m = 1, 7).

Note 2. Selected by the ICSELB bit in TPUm.TMDR (m = 0, 6).

Table 24.15 TPU1.TIOR, TPU7.TIOR

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 1, 7) Function	TIOCAn Pin (n = 1, 7) 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 TIOCAn pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCAn pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCAn pin; input capture at both edges
1	1	x	x		• TPU1 Capture input source is TPU0.TGRA compare match/input capture Input capture at generation of TPU0.TGRA compare match/input capture • TPU7 Capture input source is TPU6.TGRA compare match/input capture Input capture at generation of TPU6.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 1, 7) Function	TIOCBn Pin (n = 1, 7) 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 TIOCBn or TIOCAn pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCBn or TIOCAn pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCBn or TIOCAn pin*1; input capture at both edges
1	1	x	x		• TPU1 Capture input source is TPU0.TGRC compare match/input capture Input capture at generation of TPU0.TGRC compare match/input capture • TPU7 Capture input source is TPU6.TGRC compare match/input capture Input capture at generation of TPU6.TGRC compare match/input capture

x: Don't care
Note 1. Selected by the ICSELB bit in TPUm.TMDR (m = 1, 7).

Table 24.16 TPU2.TIOR, TPU8.TIOR

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 2, 8) Function	TIOCAn Pin (n = 2, 8) 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 TIOCAn pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCAn pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCAn pin; input capture at both edges

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 2, 8) Function	TIOCBn Pin (n = 2, 8) 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 TIOCBn or TIOCAn pin*1; input capture at rising edge
1	x	0	1		Capture input source is TIOCBn or TIOCAn pin*1; input capture at falling edge
1	x	1	x		Capture input source is TIOCBn or TIOCAn pin*1; input capture at both edges

x: Don't care
Note 1. Selected by the ICSELB bit in TPUm.TMDR (m = 2, 8).

Table 24.17 TPU3.TIORH, TPU9.TIORH

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 3, 9) Function	TIOCAn Pin (n = 3, 9) 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 TIOCAn pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCAn pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCAn pin; input capture at both edges
1	1	x	x		• TPU3 Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*1 • TPU9 Capture input source is TPU10 count clock Input capture at TPU10.TCNT count-up/count-down*1

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 3, 9) Function	TIOCBn Pin (n = 3, 9) 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 TIOCBn or TIOCAn pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCBn or TIOCAn pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCBn or TIOCAn pin*2; input capture at both edges
1	1	x	x		• TPU3 Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*1 • TPU9 Capture input source is TPU10 count clock Input capture at TPU10.TCNT count-up/count-down*1

x: Don't care

Note 1. When the TPSC[2:0] bits in TPUm.TCR are set to 000b and PCLK/1 is used as the TPUm.TCNT count clock, this setting is invalid and input capture is not generated (m = 4, 10).

Note 2. Selected by the ICSELB bit in TPUm.TMDR (m = 3, 9).

Table 24.18 TPU4.TIOR, TPU10.TIOR

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 4, 10) Function	TIOCAn Pin (n = 4, 10) 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 TIOCAn pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCAn pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCAn pin; input capture at both edges
1	1	x	x		• TPU4 Capture input source is TPU3.TGRA compare match/input capture Input capture at generation of TPU3.TGRA compare match/input capture • TPU10 Capture input source is TPU9.TGRA compare match/input capture Input capture at generation of TPU9.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 4, 10) Function	TIOCBn Pin (n = 4, 10) 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 TIOCBn or TIOCAn pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCBn or TIOCAn pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCBn or TIOCAn pin*1; input capture at both edges
1	1	x	x		• TPU4 Capture input source is TPU3.TGRC compare match/input capture Input capture at generation of TPU3.TGRC compare match/input capture • TPU10 Capture input source is TPU9.TGRC compare match/input capture Input capture at generation of TPU9.TGRC compare match/input capture

x: Don't care
Note 1. Selected by the ICSELB bit in TPUm.TMDR (m = 4, 10).

Table 24.19 TPU5.TIOR, TPU11.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPUm.TGRA (m = 5, 11) Function**	**TIOCAn Pin (n = 5, 11) 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 TIOCAn pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCAn pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCAn pin; input capture at both edges

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPUm.TGRB (m = 5, 11) Function**	**TIOCBn Pin (n = 5, 11) 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 TIOCBn or TIOCAn pin*1; input capture at rising edge
1	x	0	1		Capture input source is TIOCBn or TIOCAn pin*1; input capture at falling edge
1	x	1	x		Capture input source is TIOCBn or TIOCAn pin*1; input capture at both edges

x: Don't care
Note 1. Selected by the ICSELB bit in TPUm.TMDR (m = 5, 11).

Table 24.20 TPU0.TIORL, TPU6.TIORL

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRC (m = 0, 6) Function	TIOCCn Pin (n = 0, 6) 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 TIOCCn pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCCn pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCCn pin; input capture at both edges
1	1	x	x		• TPU0 Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*3 • TPU6 Capture input source is TPU7 count clock Input capture at TPU7.TCNT count-up/count-down*3

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRD (m = 0, 6) Function	TIOCDn Pin (n = 0, 6) 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 TIOCDn or TIOCCn pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCDn or TIOCCn pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCDn or TIOCCn pin*4; input capture at both edges
1	1	x	x		• TPU0 Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*3 • TPU6 Capture input source is TPU7 count clock Input capture at TPU7.TCNT count-up/count-down*3

x: Don't care

Note 1. When the BFA bit in TPUm.TMDR is set to 1 (TPUm.TGRA and TPUm.TGRC are used for buffer operation) and TPUm.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated (m = 0, 6).

Note 2. When the BFB bit in TPUm.TMDR is set to 1 (TPUm.TGRB and TPUm.TGRD are used for buffer operation) and TPUm.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated (m = 0, 6).

Note 3. When the TPSC[2:0] bits in TPUm.TCR are set to 000b and PCLK/1 is used as the TPUm.TCNT count clock, this setting is invalid and input capture is not generated (m = 1, 7).

Note 4. Selected by the ICSELD bit in TPUm.TMDR (m = 0, 6).

Table 24.21 TPU3.TIORL, TPU9.TIORL

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRC (m = 3, 9) Function	TIOCCn Pin (n = 3, 9) 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 TIOCCn pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCCn pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCCn pin; input capture at both edges
1	1	x	x		• TPU3 Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*3 • TPU9 Capture input source is TPU10 count clock Input capture at TPU10.TCNT count-up/count-down*3

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRD (m = 3, 9) Function	TIOCDn Pin (n = 3, 9) 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 TIOCDn or TIOCCn pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCDn or TIOCCn pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCDn or TIOCCn pin*4; input capture at both edges
1	1	X	x		• TPU3 Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*3 • TPU9 Capture input source is TPU10 count clock Input capture at TPU10.TCNT count-up/count-down*3

x: Don't care

Note 1. When the BFA bit in TPUm.TMDR is set to 1 (TPUm.TGRA and TPUm.TGRC are used for buffer operation) and TPUm.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated (m = 3, 9).

Note 2. When the BFB bit in TPUm.TMDR is set to 1 (TPUm.TGRB and TPUm.TGRD are used for buffer operation) and TPUm.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated (m = 3, 9).

Note 3. When the TPSC[2:0] bits in TPUm.TCR are set to 000b and PCLK/1 is used as the TPUm.TCNT count clock, this setting is invalid and input capture is not generated (m = 4, 10).

Note 4. Selected by the ICSELD bit in TPUm.TMDR (m = 3, 9).

24.2.4 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, TPU6.TIER 0008 8184h, TPU7.TIER 0008 8194h, TPU8.TIER 0008 81A4h, TPU9.TIER 0008 81B4h, TPU10.TIER 0008 81C4h, TPU11.TIER 0008 81D4h

	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 11)	R/W
b1	TGIEB	TGRB Interrupt Enable	0: Interrupt requests (TGImB) disabled 1: Interrupt requests (TGImB) enabled (m = 0 to 11)	R/W
b2	TGIEC*1	TGRC Interrupt Enable	0: Interrupt requests (TGImC) disabled 1: Interrupt requests (TGImC) enabled (m = 0, 3, 6, 9)	R/W
b3	TGIED*1	TGRD Interrupt Enable	0: Interrupt requests (TGImD) disabled 1: Interrupt requests (TGImD) enabled (m = 0, 3, 6, 9)	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCImV) disabled 1: Interrupt requests (TCImV) enabled (m = 0 to 11)	R/W
b5	TCIEU*2	Underflow Interrupt Enable	0: Interrupt requests (TCImU) disabled 1: Interrupt requests (TCImU) enabled (m = 1, 2, 4, 5, 7, 8, 10, 11)	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TTGE	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, TPU5 (unit 0), TPU7, TPU8, TPU10, and TPU11 (unit 1) are reserved. These bits are read as 0. The write value should be 0.
Note 2. Bit 5 in TIER of TPU0, TPU3 (unit 0), TPU6, and TPU9 (unit 1) 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 11) input capture/compare match.

24.2.5 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, TPU6.TSR 0008 8185h, TPU7.TSR 0008 8195h, TPU8.TSR 0008 81A5h, TPU9.TSR 0008 81B5h, TPU10.TSR 0008 81C5h, TPU11.TSR 0008 81D5h

	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 11)	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 11)	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, 6, 9)	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, 6, 9)	R/W*2
b4	TCFV	Overflow Flag	0: TPUm.TCNT has not overflowed. 1: TPUm.TCNT has overflowed. (m = 0 to 11)	R/W*2
b5	TCFU*3	Underflow Flag	0: TPUm.TCNT has not underflowed. 1: TPUm.TCNT has underflowed. (m = 1, 2, 4, 5, 7, 8, 10, 11)	R/W*2
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R
b7	TCFD*1	Counting Direction Flag	0: Counter (TPUm.TCNT) counts down. 1: Counter (TPUm.TCNT) counts up. (m = 1, 2, 4, 5, 7, 8, 10, 11)	R

Note 1. Bit 7 of registers TPU0.TSR, TPU3.TSR, TPU6.TSR, and TPU9.TSR is reserved. The bit is read as 1. When writing, write 1 to this bit.
Note 2. Only writing 0 to this bit is possible; this clears the flag.
Note 3. Bit 5 of registers TPU0.TSR, TPU3.TSR, TPU6.TSR, and TPU9.TSR is reserved. The bit is read as 0. When writing, write 0 to this bit.
Note 4. Bits 2 and 3 of registers TPU1.TSR, TPU2.TSR, TPU4.TSR, TPU5.TSR, TPU7.TSR, TPU8.TSR TPU10.TSR, and TPU11.TSR are reserved. The bits are read as 0. When writing, write 0 to these bits.

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 11) 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 11) 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, 6, 9) 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, 6, 9) 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 11).

[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, 7, 8, 10, 11).

[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.

24.2.6 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, TPU6.TCNT 0008 8186h, TPU7.TCNT 0008 8196h, TPU8.TCNT 0008 81A6h, TPU9.TCNT 0008 81B6h, TPU10.TCNT 0008 81C6h, TPU11.TCNT 0008 81D6h

	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 16-bit counter that counts the internal clock or external events.

This counter can be read/written in 16-bit units.

This counter is initialized to 0000h by a reset.

24.2.7 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,
TPU6.TGRA 0008 8188h, TPU6.TGRB 0008 818Ah, TPU6.TGRC 0008 818Ch, TPU6.TGRD 0008 818Eh,
TPU7.TGRA 0008 8198h, TPU7.TGRB 0008 819Ah,
TPU8.TGRA 0008 81A8h, TPU8.TGRB 0008 81AAh,
TPU9.TGRA 0008 81B8h, TPU9.TGRB 0008 81BAh, TPU9.TGRC 0008 81BCh, TPU9.TGRD 0008 81BEh,
TPU10.TGRA 0008 81C8h, TPU10.TGRB 0008 81CAh,
TPU11.TGRA 0008 81D8h, TPU11.TGRB 0008 81DAh

	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 32 TGR registers in total, four each for TPU0, TPU3, TPU6, and TPU9 and two each for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11.

TPUm.TGRA (m = 0 to 11), TPUm.TGRB (m = 0 to 11), TPUm.TGRC (m = 0, 3, 6, 9), and TPUm.TGRD (m = 0, 3, 6, 9) are 16-bit registers with a dual function as output compare and input capture registers.

These registers can be read/written in 16-bit units.

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.

24.2.8 Timer Start Register (TSTR)

- Unit 0 (TPUA.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

- Unit 1 (TPUB.TSTR)

Address(es): TPUB.TSTR 0008 8170h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CST11	CST10	CST9	CST8	CST7	CST6
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST6	Counter Start 6	0: TCNT count operation is stopped 1: TCNT performs count operation	R/W
b1	CST7	Counter Start 7		R/W
b2	CST8	Counter Start 8		R/W
b3	CST9	Counter Start 9		R/W
b4	CST10	Counter Start 10		R/W
b5	CST11	Counter Start 11		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TSTRA starts or stops TCNT count operation for TPU0 to TPU5.
TSTRB starts or stops TCNT count operation for TPU6 to TPU11.
Before setting the operating mode in TPUm.TMDR or setting the TPUm.TCNT count clock in TPUm.TCR, stop the TPUm.TCNT counter.

CSTn Bits (Counter Start) (n = 0 to 11)

These bits start or stop the TCNT counter.
When the CSTn bit is cleared to 0 with CSTn = 1 and the corresponding TIOCyn pin (y = A to D, n = 0 to 11) specified for output, the counter 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.

24.2.9 Timer Synchronous Register (TSYR)

- Unit 0 (TPUA.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 presetting/clearing is unrelated to other channels) 1: TCNT performs synchronous operation*1 (TCNT synchronous presetting/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 CCLR[2:0] bits in TCR in addition to the SYNCn bit.

- Unit 1 (TPUB.TSYR)

Address(es): TPUB.TSYR 0008 8171h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SYNC11	SYNC10	SYNC9	SYNC8	SYNC7	SYNC6
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC6	Timer Synchronization 6	0: TCNT operates independently (TCNT presetting/clearing is unrelated to other channels) 1: TCNT performs synchronous operation*1 (TCNT synchronous presetting/synchronous clearing is possible)	R/W
b1	SYNC7	Timer Synchronization 7		R/W
b2	SYNC8	Timer Synchronization 8		R/W
b3	SYNC9	Timer Synchronization 9		R/W
b4	SYNC10	Timer Synchronization 10		R/W
b5	SYNC11	Timer Synchronization 11		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 = 6 to 11) for at least two channels must be set to 1. To set synchronous clearing, the TCNT clearing source must also be set by the CCLR[2:0] bits in TCR in addition to the SYNCn bit.

TPUA.TSYR selects independent operation or synchronous operation for the TCNT counters of TPU0 to TPU5.
TPUB.TSYR selects independent operation or synchronous operation for the TCNT counters of TPU6 to TPU11.

SYNCn Bits (Timer Synchronization) (n = 0 to 11)

These bits select whether the TCNT operation is independent of or synchronized with TCNT of other channels.
When synchronous operation is selected, synchronous presetting of multiple TCNT counters and synchronous clearing through counter clearing on another channel are possible.

24.2.10 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, TPU6.NFCR 0008 8178h, TPU7.NFCR 0008 8179h, TPU8.NFCR 0008 817Ah, TPU9.NFCR 0008 817Bh, TPU10.NFCR 0008 817Ch, TPU11.NFCR 0008 817Dh

	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 11)	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 11)	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, 6, 9)	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, 6, 9)	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, TPU5.NFCR, TPU7.NFCR, TPU8.NFCR, TPU10.NFCR, and TPU11.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 counter is stopped.

NFAEN Bit (Noise Filter Enable A)

This bit disables or enables the noise filter for the TIOCAm pin (m = 0 to 11).

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 11).

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, 6, 9).

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, 6, 9).
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 counter 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.

24.3 Operation

24.3.1 Basic Functions

Each channel has a TPUm.TCNT counter 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 or the CSTj bit (j = 6 to 11) in TPUB.TSTR is set to 1, the TCNT counter for the corresponding channel starts counting.

(a) Example of count operation setting procedure

Figure 24.3 shows an example of the count operation setting procedure.

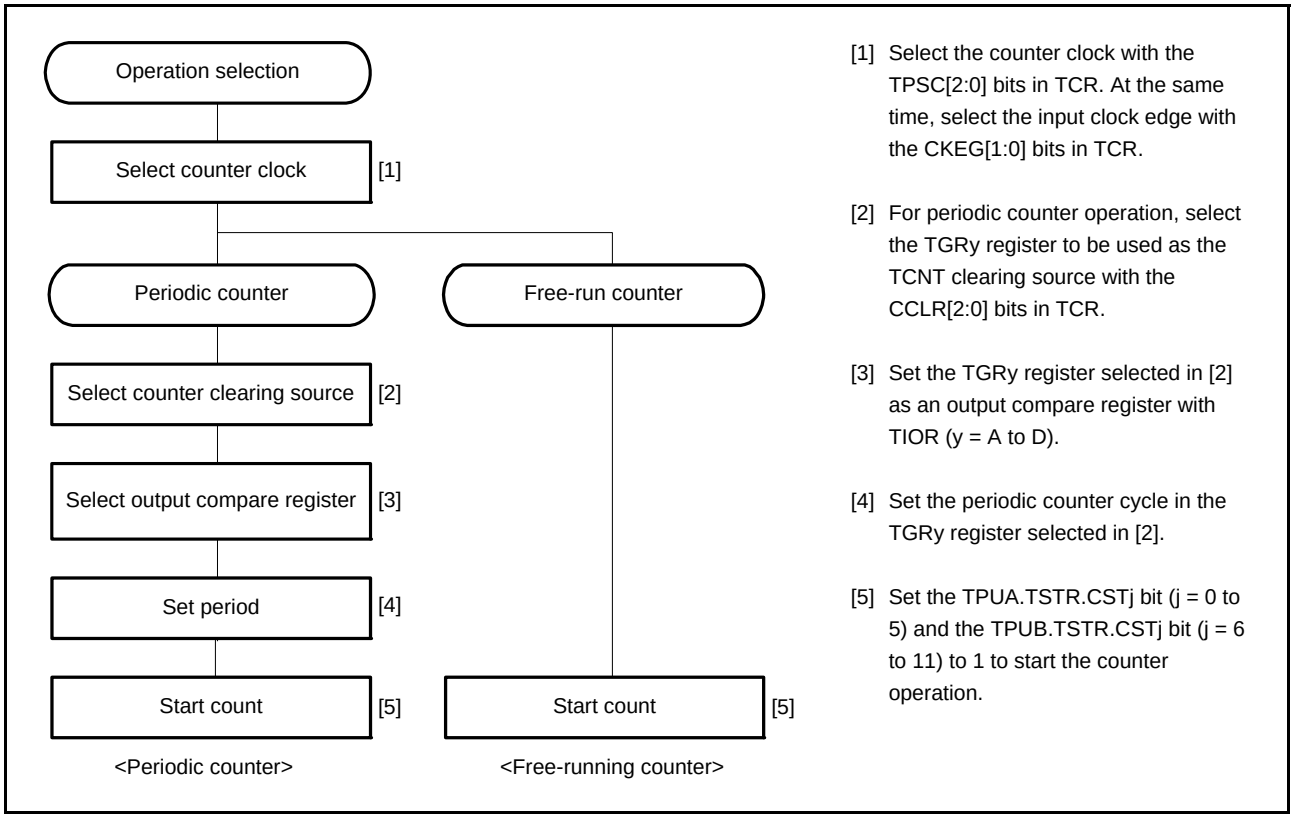


Figure 24.3 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, the TPUm.TCNT counters are all set as free-running counters. When the relevant bit in TPUA.TSTR or TPUB.TSTR is set to 1, the corresponding TCNT counter 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 24.4 shows free-running counter operation.

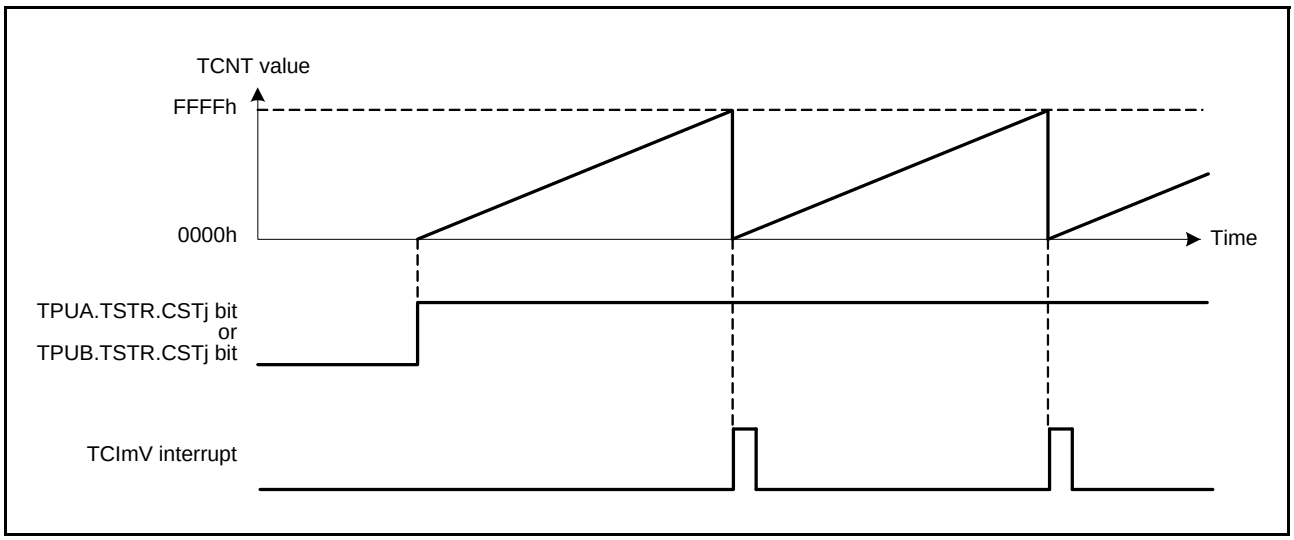


Figure 24.4 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter 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 CCLR[2:0] bits in TPUm.TCR. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TPUA.TSTR or TPUB.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 24.5 shows periodic counter operation.

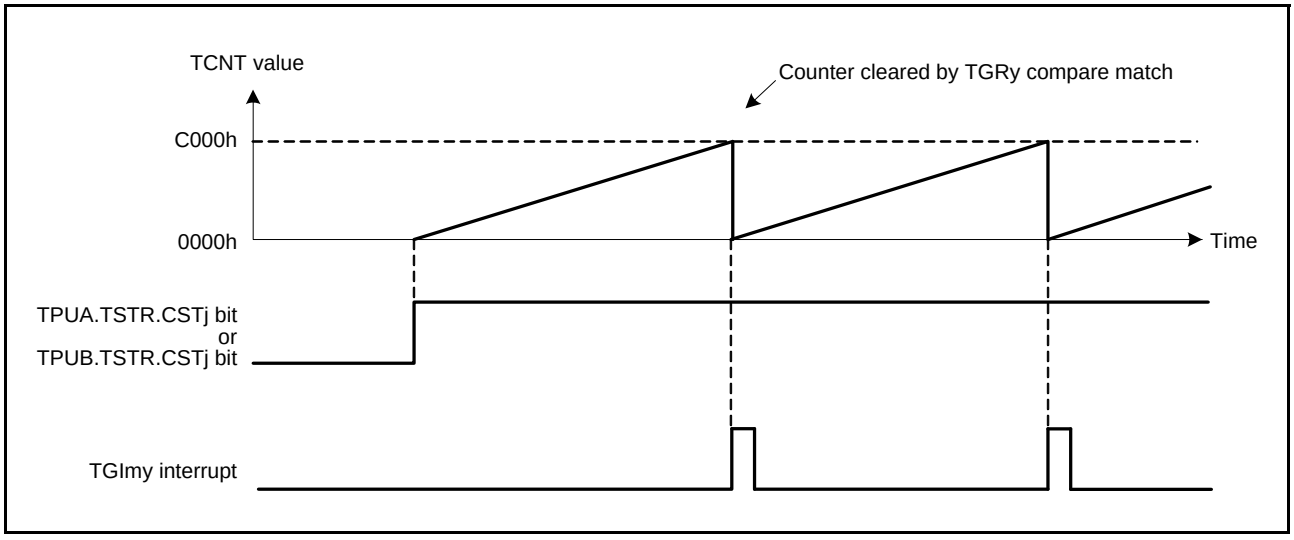


Figure 24.5 Periodic Counter Operation

(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 24.6 shows an example of the setting procedure for waveform output by a compare match.

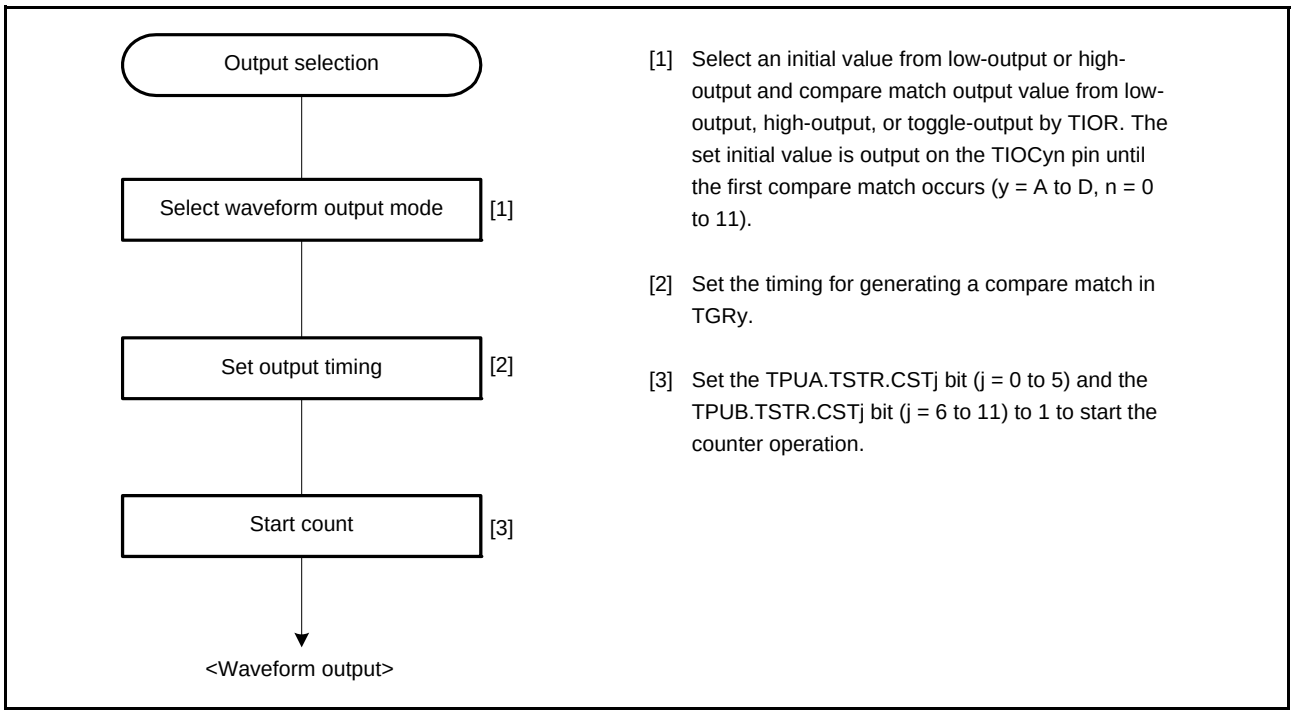


Figure 24.6 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 24.7 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.

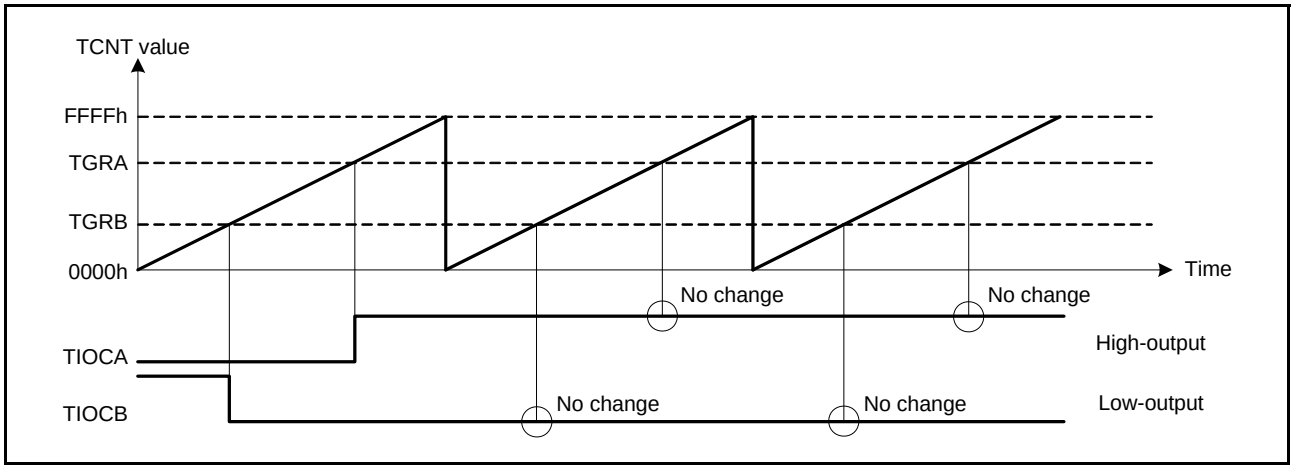


Figure 24.7 Example of Low-Output/High-Output Operation

Figure 24.8 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.

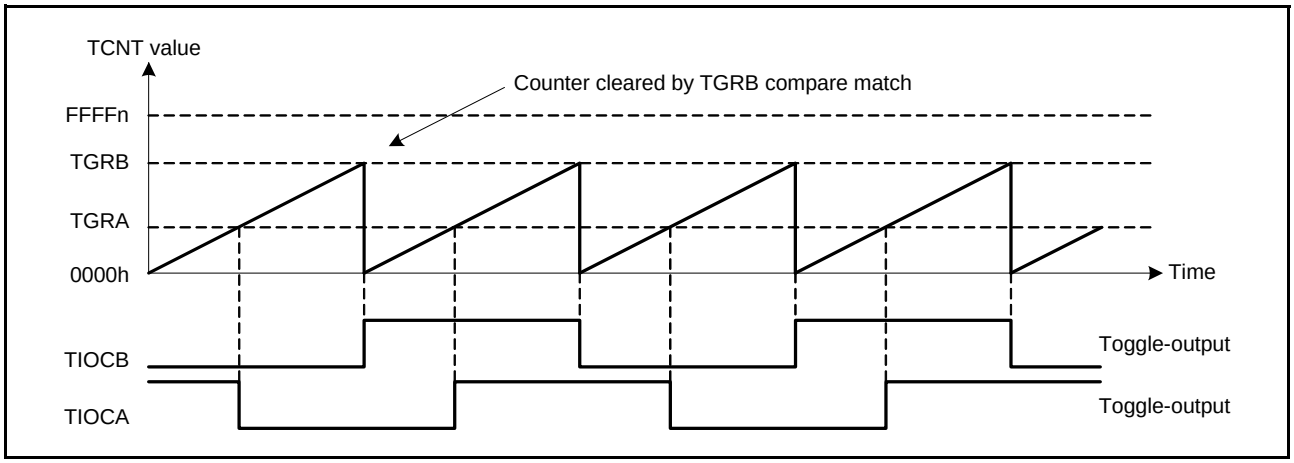


Figure 24.8 Example of Toggle Output Operation

(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 11) 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 counter input clock or compare match signal of TPU0, TPU1, TPU3, and TPU4 (TPU6, TPU7, TPU9, and TPU10) as the input capture source. Noise filtering can be applied to the input capture input.

Note 1. • Even if the counter is halted, an input capture is generated, and flag and interrupt signals are generated.
. • When another channel's counter input clock is used as the input capture input for TPU0 and TPU3 (TPU6 and TPU9), PCLK/1 should not be selected as the counter input 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 24.9 shows an example of the setting procedure for input capture operation.

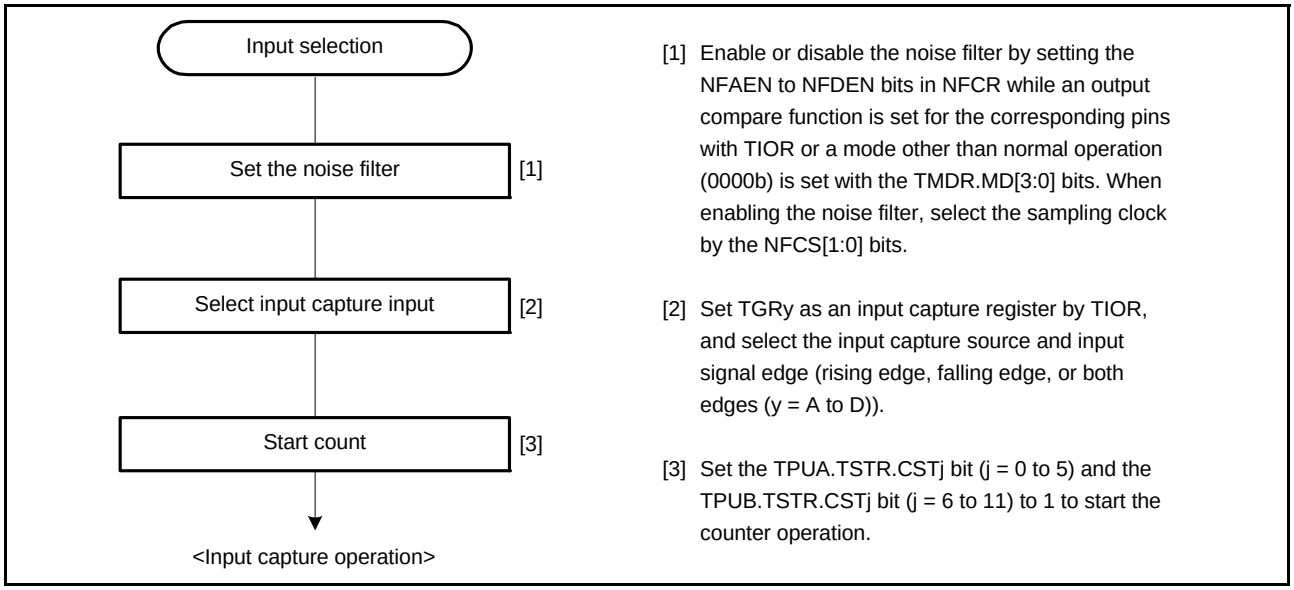


Figure 24.9 Example of Setting Procedure for Input Capture Operation

(b) Example of input capture operation

Figure 24.10 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.

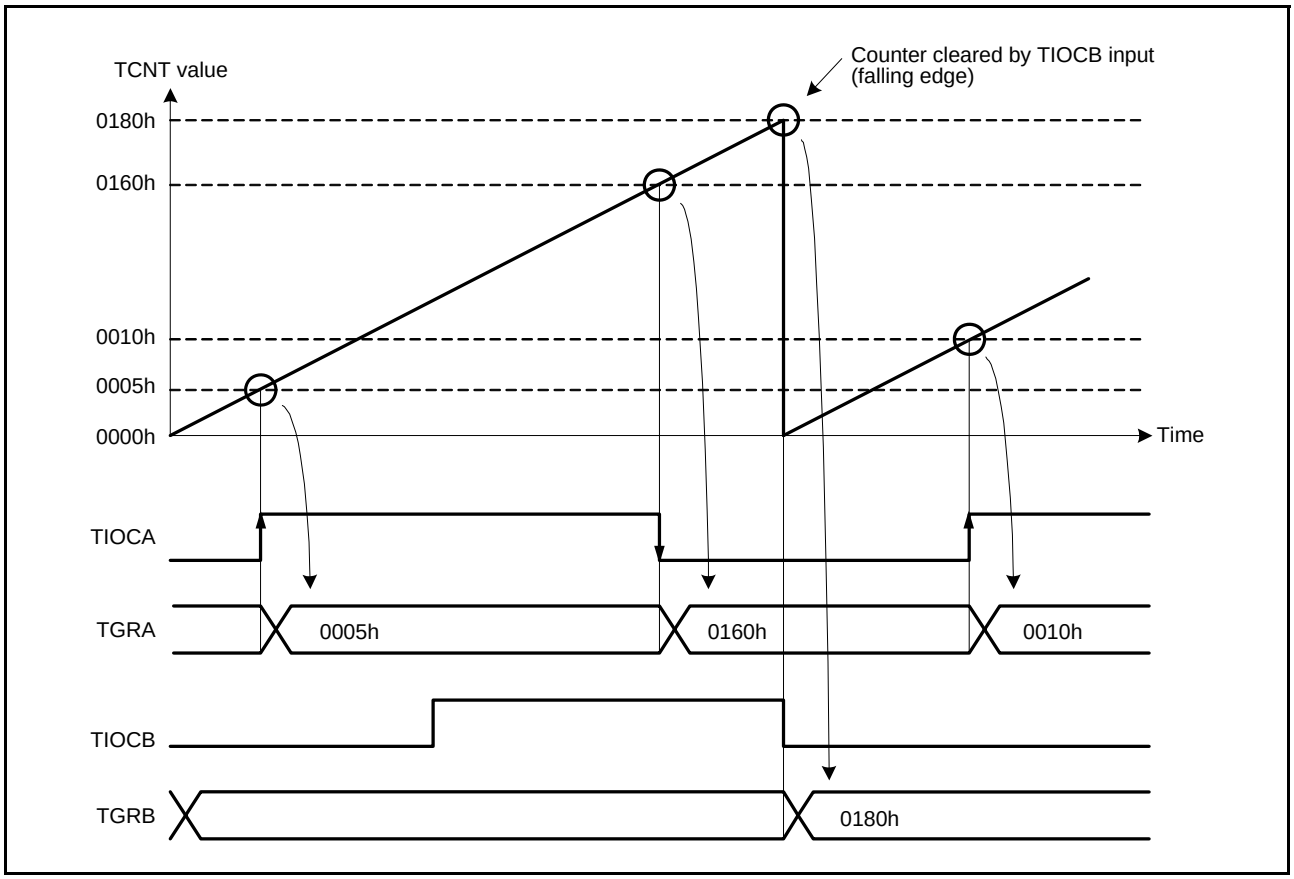


Figure 24.10 Example of Input Capture Operation (with Noise Filter Stopped)

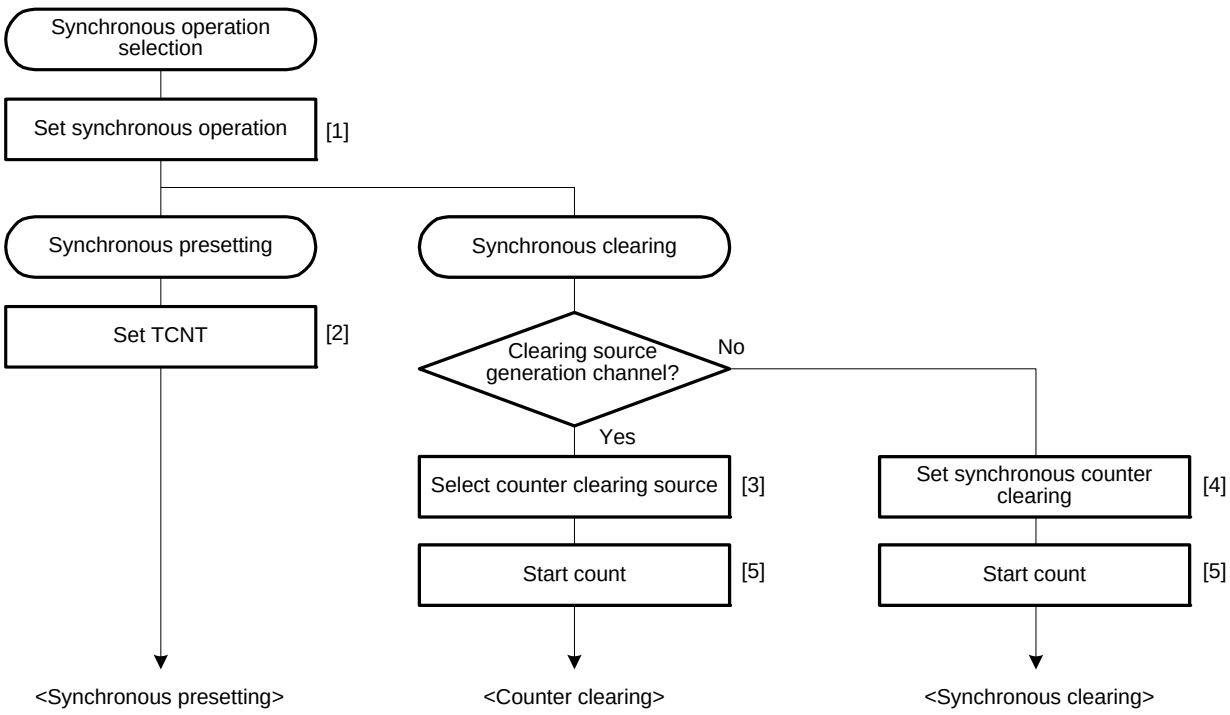
If noise filtering is enabled, input capture operation is performed on the edges of noise-filtered signal after a delay of (minimum sampling interval × 2 + PCLK) due to noise filtering for the input capture input.

24.3.2 Synchronous Operation

In synchronous operation, the values in multiple TPUm.TCNT counters can be rewritten simultaneously (synchronous presetting). Also, multiple TCNT counters 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 (TPU6 to TPU11) can all be set for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 24.11 shows an example of the synchronous operation setting procedure.



[1] Set the TPUA.TSYR.SYNCj bit (j = 0 to 5) and the TPUB.TSYR.SYNCj bit (j = 6 to 11) corresponding to the channels to be set for synchronous operation to 1.

[2] When any of TCNT of the channels set 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 specify synchronous clearing for the counter clearing source.

[5] Set the TPUA.TSTR.CSTj bit (j = 0 to 5) and the TPUB.TSTR.CSTj bit (j = 6 to 11) corresponding to the target channel to 1 to start the counter operation.

Figure 24.11 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 24.12 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 presetting 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 24.3.5, PWM Modes.

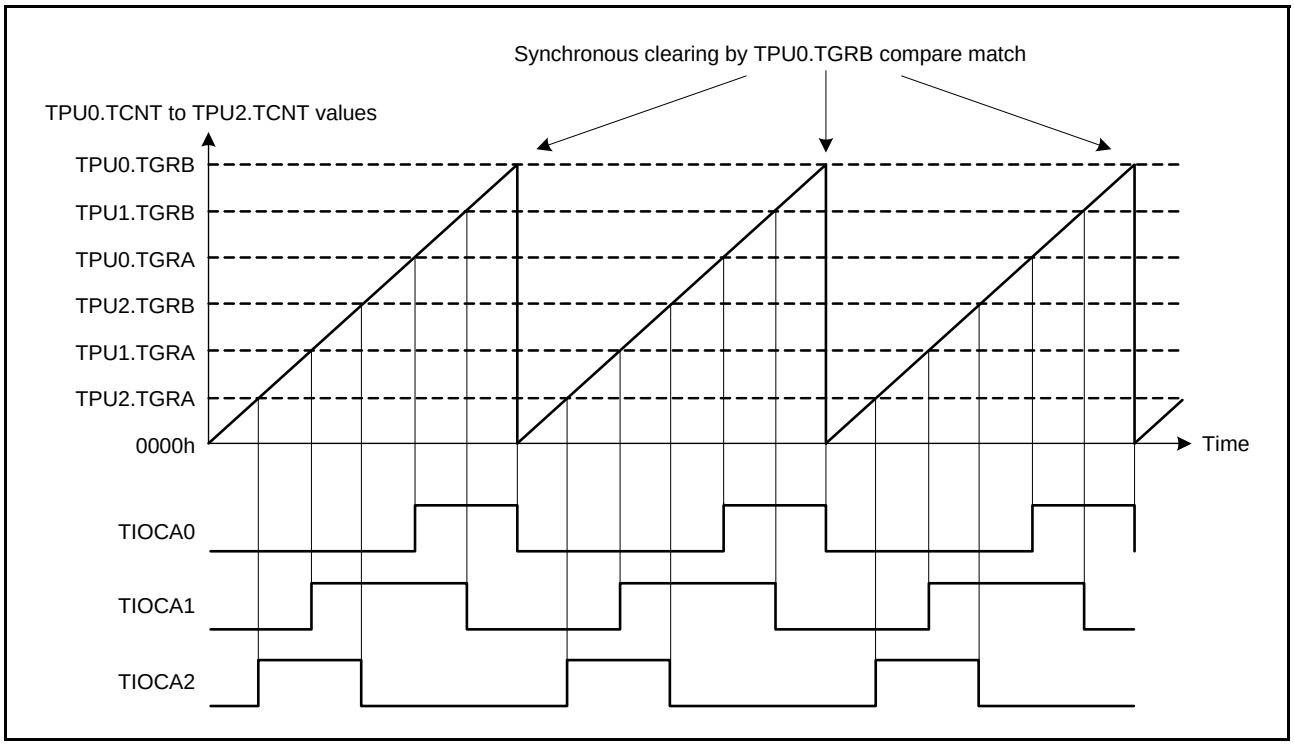


Figure 24.12 Example of Synchronous Operation

24.3.3 Buffer Operation

Buffer operation, provided for TPU0 and TPU3 (TPU6 and TPU9), 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 24.22 lists the register combinations used in buffer operation.

Table 24.22 Register Combinations in Buffer Operation

Unit	Channel	Timer General Register	Buffer Register
0	TPU0	TPU0.TGRA	TPU0.TGRC
		TPU0.TGRB	TPU0.TGRD
	TPU3	TPU3.TGRA	TPU3.TGRC
		TPU3.TGRB	TPU3.TGRD
1	TPU6	TPU6.TGRA	TPU6.TGRC
		TPU6.TGRB	TPU6.TGRD
	TPU9	TPU9.TGRA	TPU9.TGRC
		TPU9.TGRB	TPU9.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 24.13.

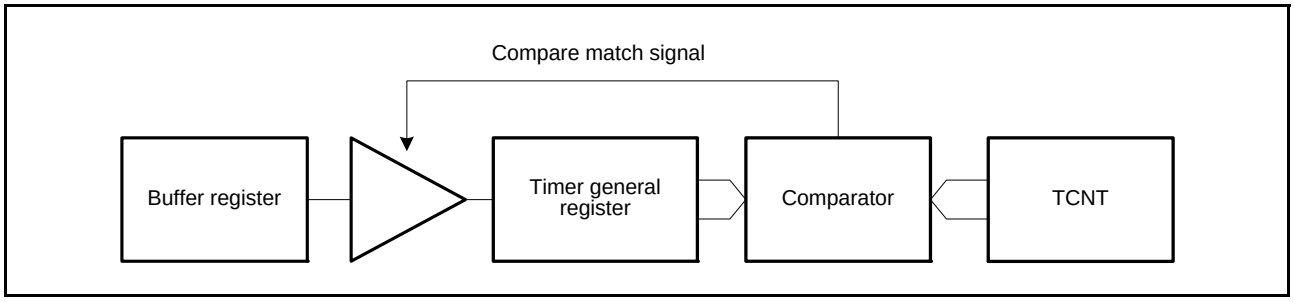


Figure 24.13 Compare Match Buffer Operation

- 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 24.14.

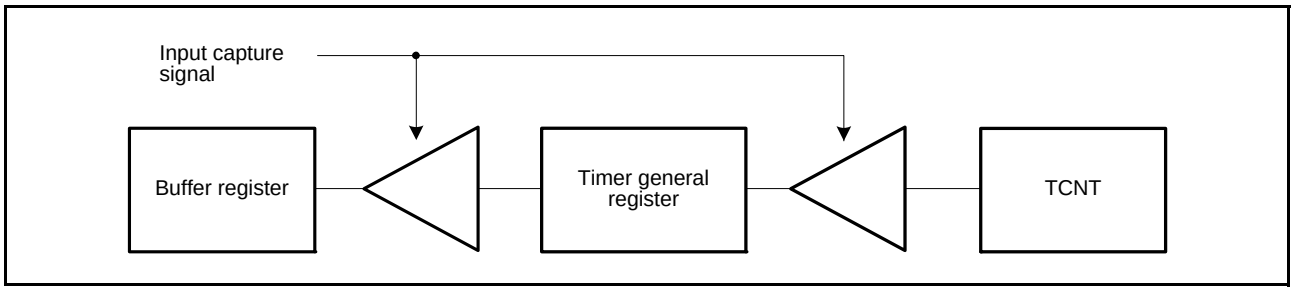


Figure 24.14 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 24.15 shows an example of the buffer operation setting procedure.

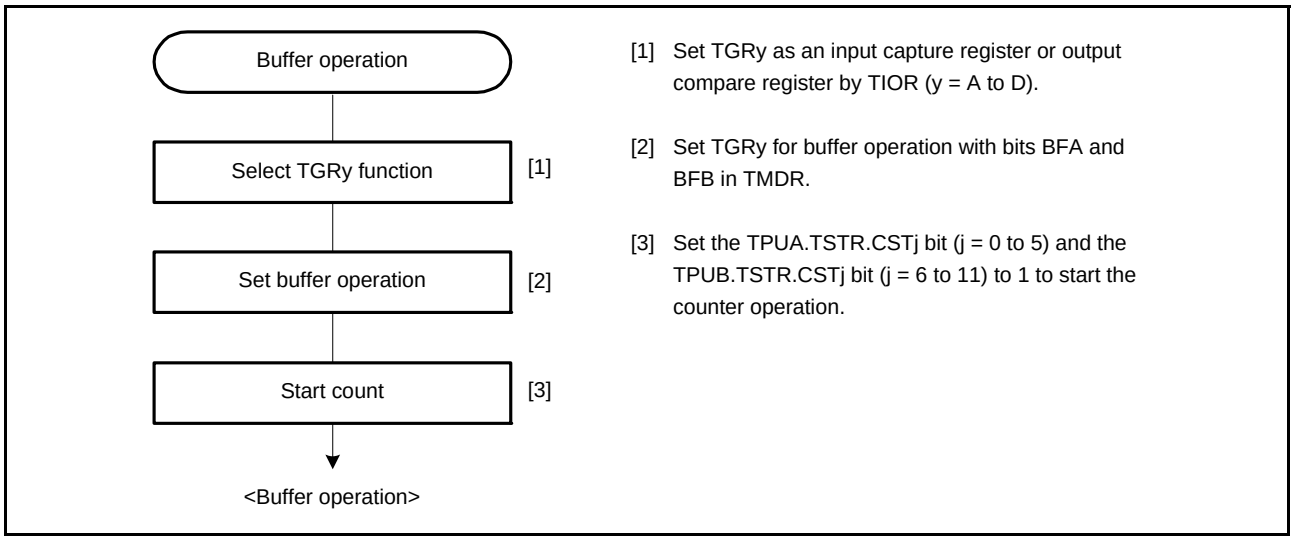


Figure 24.15 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TPUm.TGRy is an output compare register

Figure 24.16 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 24.3.5, PWM Modes.

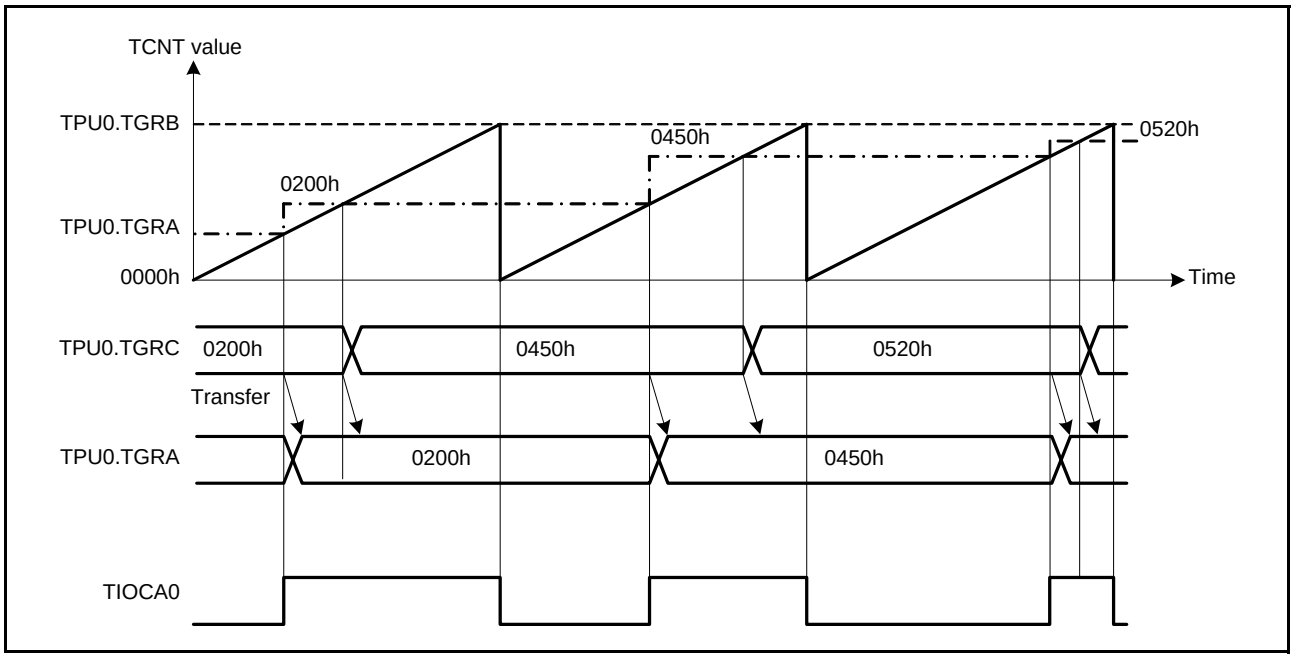


Figure 24.16 Example of Buffer Operation (1)

(b) When TPUm.TGRy is an input capture register

Figure 24.17 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.

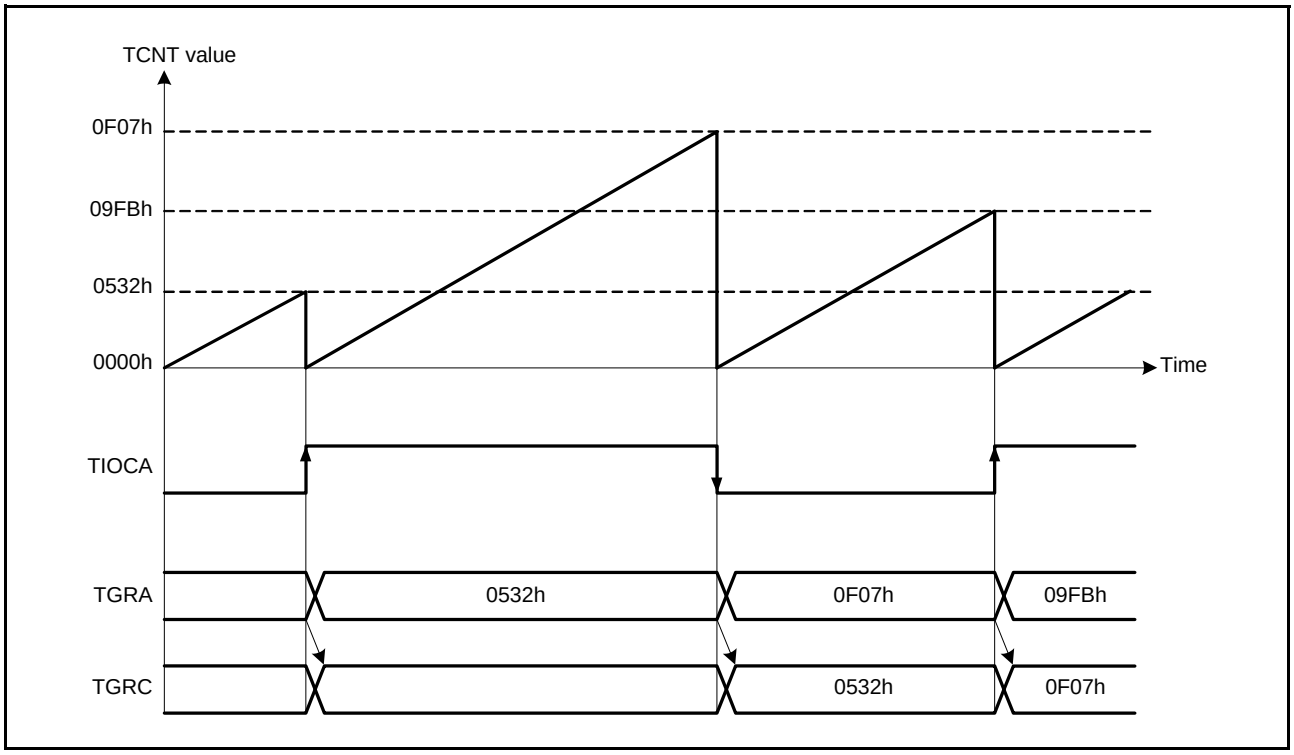


Figure 24.17 Example of Buffer Operation (2)

24.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.
In the case of unit 0, this function works by counting the TPU1 (TPU4) counter 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).
In the case of unit 1, this function works by counting the TPU7 (TPU10) counter clock at overflow/underflow of TPU8.TCNT (TPU11.TCNT) as set by the TPSC[2:0] bits in TPU7.TCR (TPSC[2:0] bits in TPU10.TCR).
Underflow occurs only when the lower 16-bit TPUm.TCNT is in phase counting mode.
Table 24.23 lists the register combinations used in cascaded operation.

Note 1. When phase counting mode is set for TPU1 or TPU4 (TPU7 or TPU10), the counter clock setting is invalid and the counter operates independently in phase counting mode.

Table 24.23 Cascaded Combinations

Unit	Combination	Upper 16 Bits	Lower 16 Bits
0	TPU1 and TPU2	TPU1.TCNT	TPU2.TCNT
	TPU4 and TPU5	TPU4.TCNT	TPU5.TCNT
1	TPU7 and TPU8	TPU7.TCNT	TPU8.TCNT
	TPU10 and TPU11	TPU10.TCNT	TPU11.TCNT

(1) Example of Cascaded Operation Setting Procedure

Figure 24.18 shows an example of the setting procedure for cascaded operation.

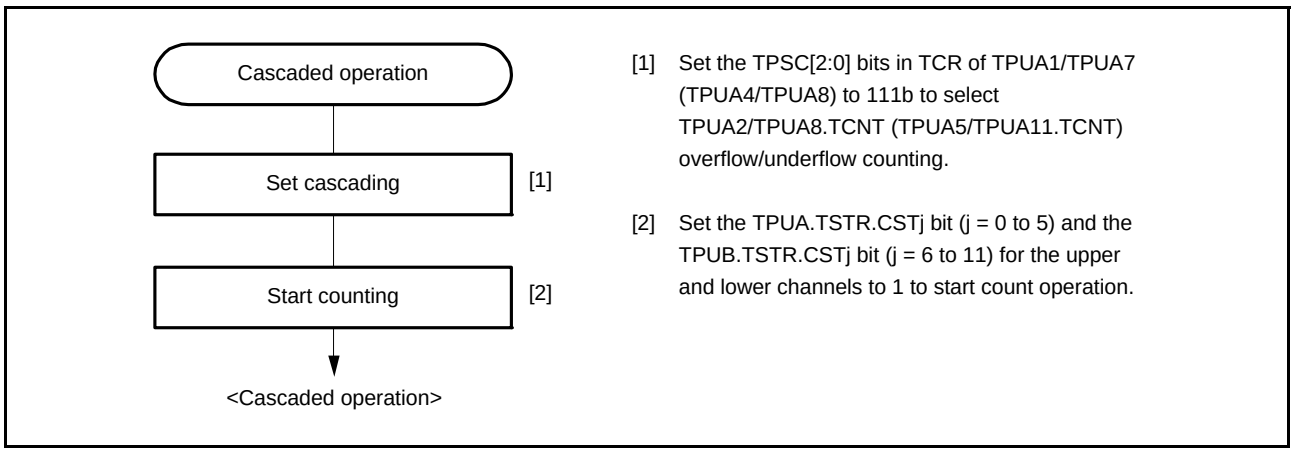


Figure 24.18 Cascaded Operation Setting Procedure

(2) Examples of Cascaded Operation

Figure 24.19 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.

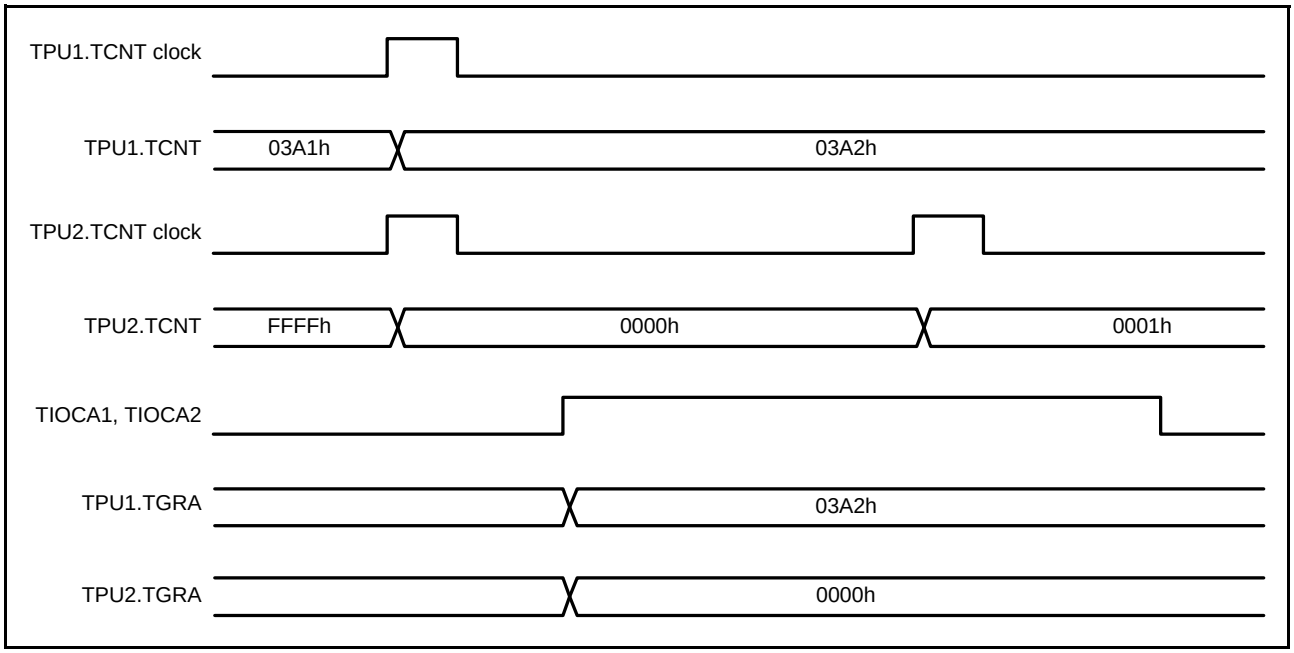


Figure 24.19 Example of Cascaded Operation (1)

Figure 24.20 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, and phase counting mode has been specified for TPU2.
TPU1.TCNT is incremented by TPU2.TCNT overflow and decremented by TPU2.TCNT underflow.

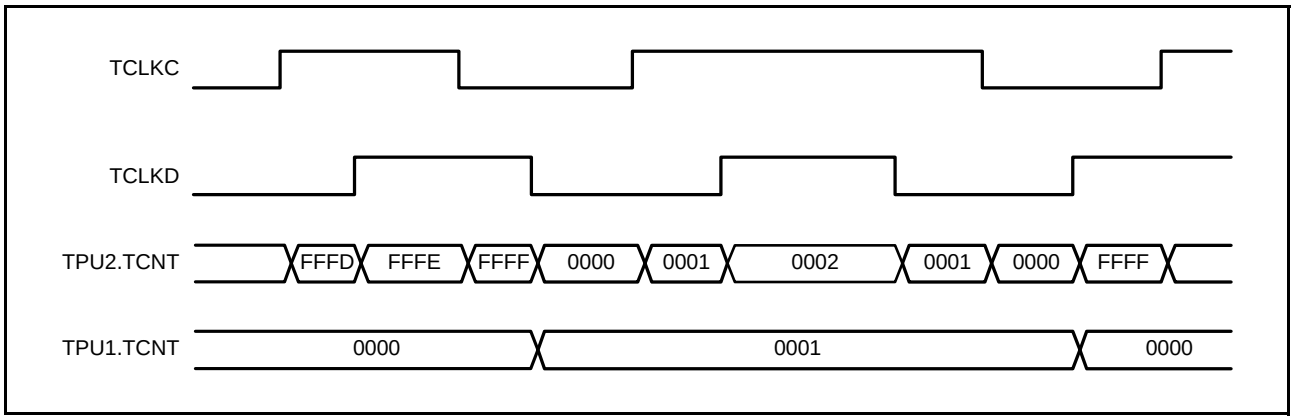


Figure 24.20 Example of Cascaded Operation (2)

24.3.5 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 24.24.

Table 24.24 PWM Output Registers and Output Pins

Unit	Channel	Register	Output Pin	
			PWM Mode 1	PWM Mode 2
0	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
1	TPU6	TPU6.TGRA	TIOCA6	TIOCA6
		TPU6.TGRB		TIOCB6
		TPU6.TGRC	TIOCC6	TIOCC6
		TPU6.TGRD		TIOCD6
	TPU7	TPU7.TGRA	TIOCA7	TIOCA7
		TPU7.TGRB		TIOCB7
	TPU8	TPU8.TGRA	TIOCA8	TIOCA8
		TPU8.TGRB		TIOCB8
	TPU9	TPU9.TGRA	TIOCA9	TIOCA9
		TPU9.TGRB		TIOCB9
		TPU9.TGRC	TIOCC9	TIOCC9
		TPU9.TGRD		TIOCD9
	TPU10	TPU10.TGRA	TIOCA10	TIOCA10
		TPU10.TGRB		TIOCB10
	TPU11	TPU11.TGRA	TIOCA11	TIOCA11
		TPU11.TGRB		TIOCB11

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 24.21 shows an example of the PWM mode setting procedure.

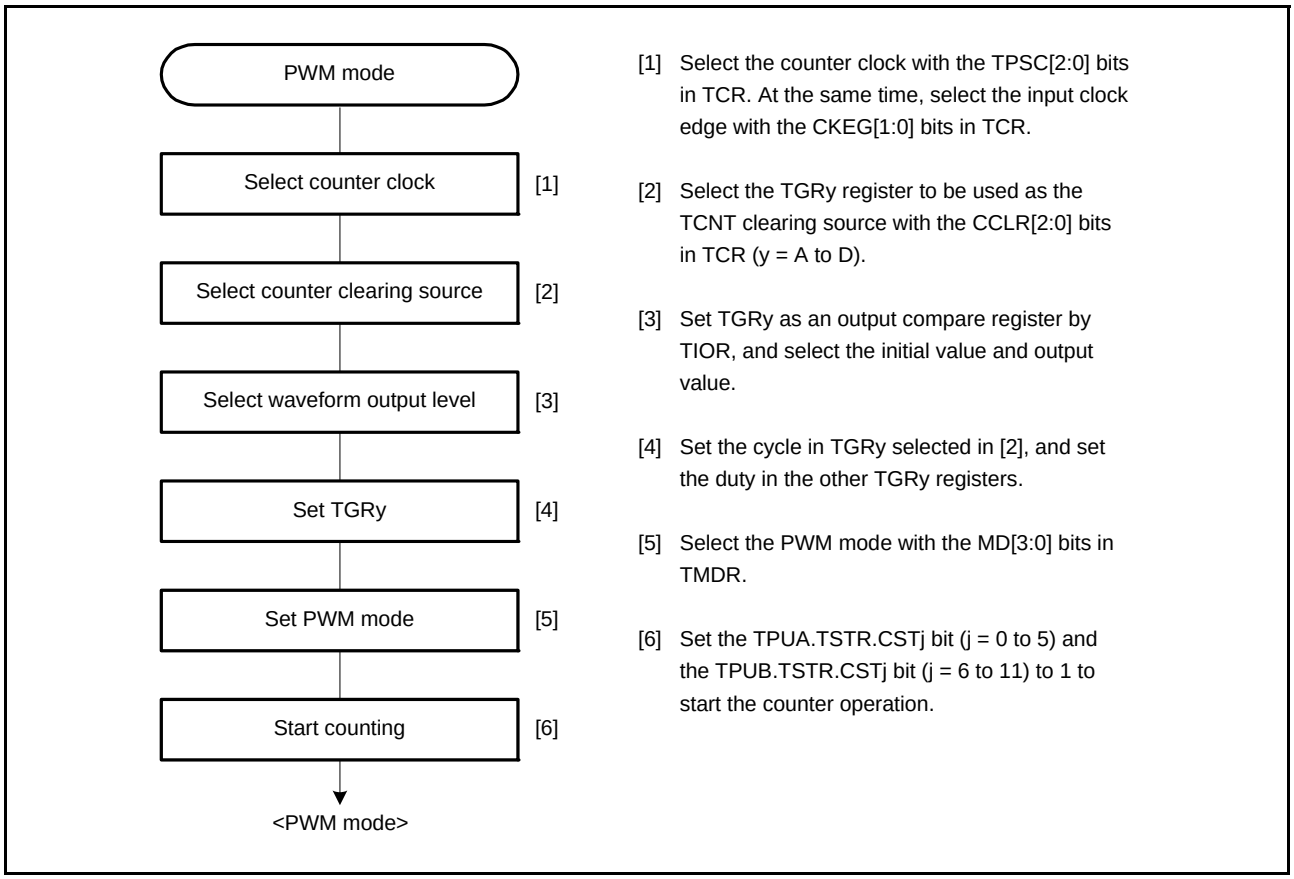


Figure 24.21 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 24.22 shows an example of PWM mode 1 operation.
In this example, TPUm.TGRA compare match is set as the TPUm.TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 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.

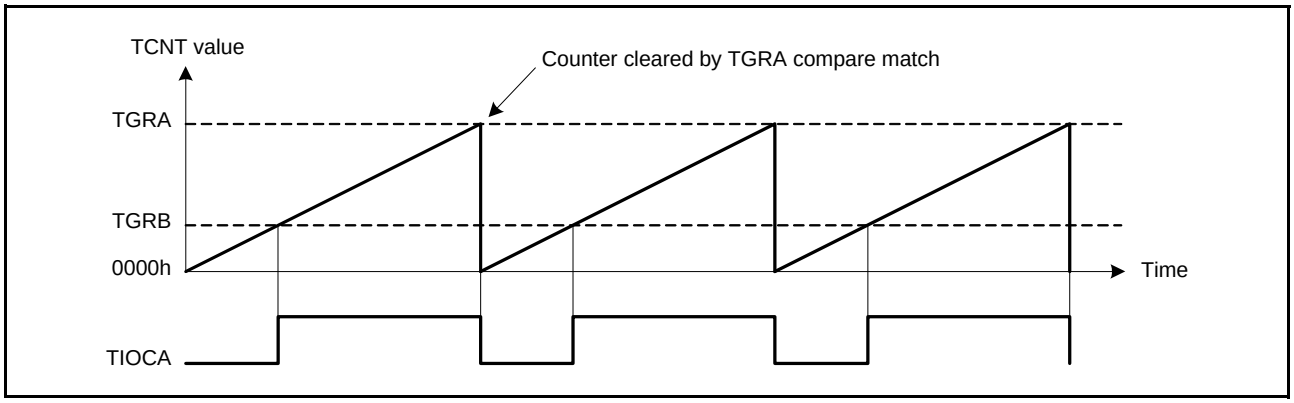


Figure 24.22 Example of PWM Mode Operation (1)

Figure 24.23 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 0 is set for the initial output value and 1 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.

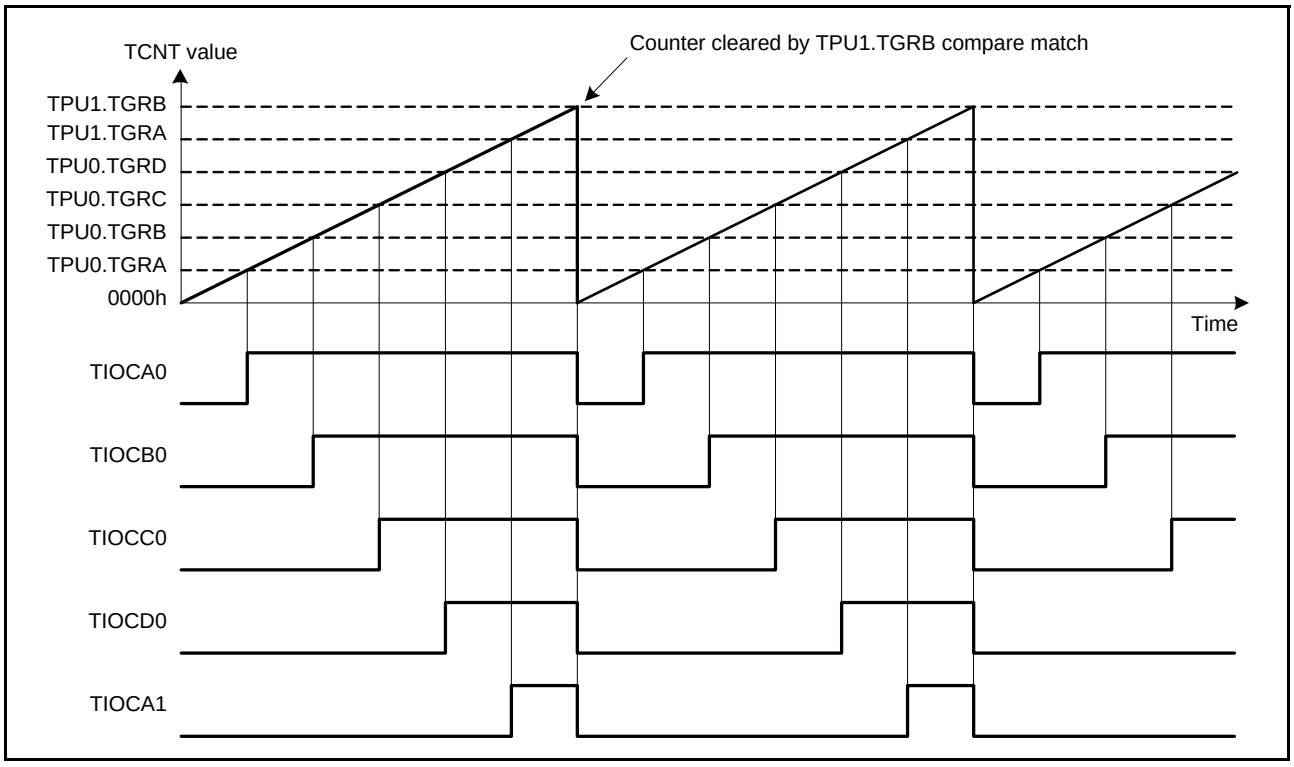


Figure 24.23 Example of PWM Mode Operation (2)

Figure 24.24 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

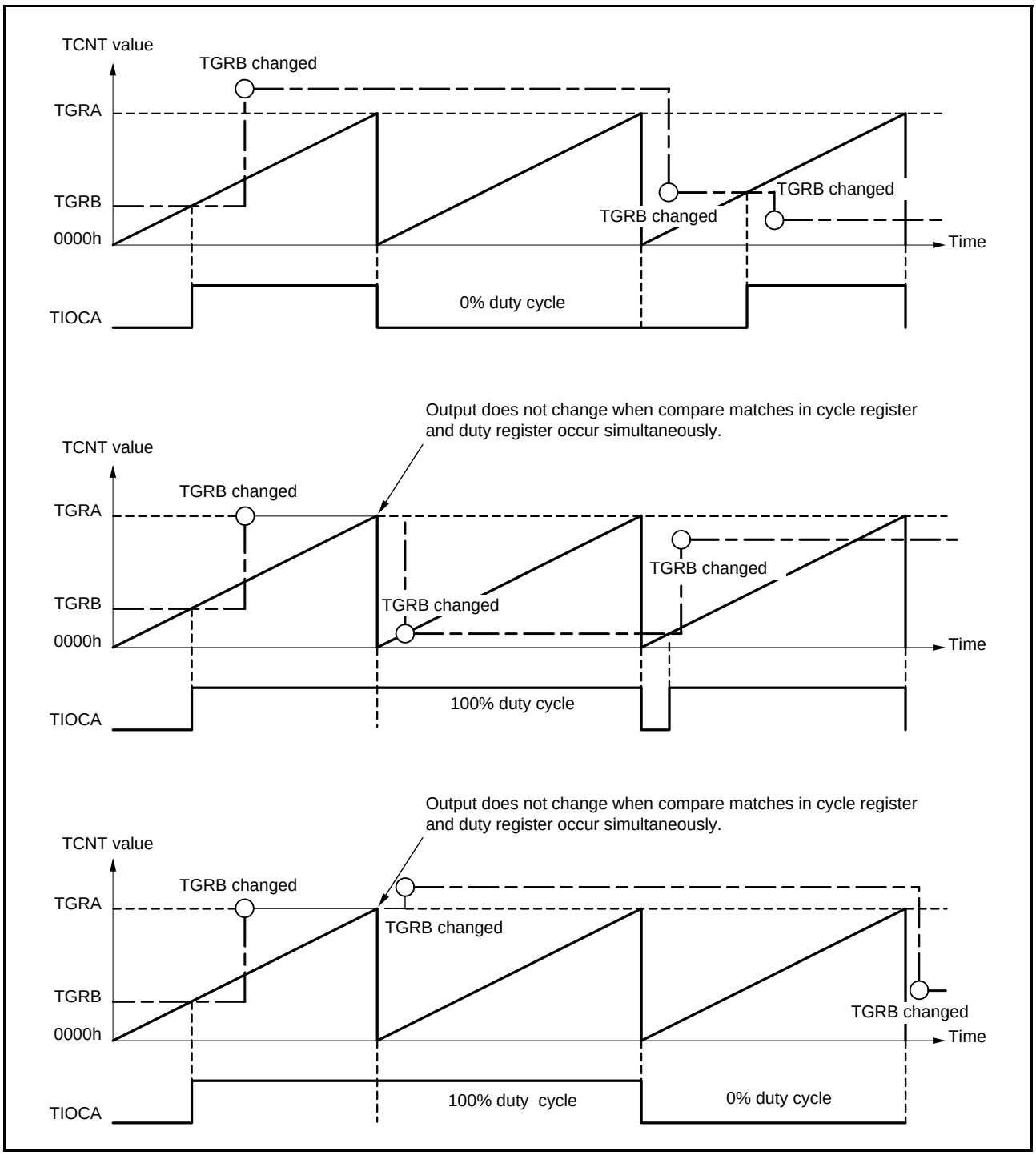


Figure 24.24 Example of PWM Mode Operation (3)

24.3.6 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 (unit 0) and channels 7, 8, 10, and 11 (unit 1), and TPUm.TCNT is incremented/decremented accordingly. When phase counting mode is set, an external clock is selected as the counter input 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.
This can be used for two-phase encoder pulse input.
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.
Table 24.25 lists the correspondence between external clock pins and channels.

Table 24.25 Clock Input Pins in Phase Counting Mode

		External Clock Pins	
Unit	**Channel**	**A-Phase**	**B-Phase**
0	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	When TPU7 or TPU11 is set to phase counting mode	TCLKE	TCLKF
	When TPU8 or TPU10 is set to phase counting mode	TCLKG	TCLKH

(1) Example of Phase Counting Mode Setting Procedure

Figure 24.25 shows an example of the phase counting mode setting procedure.

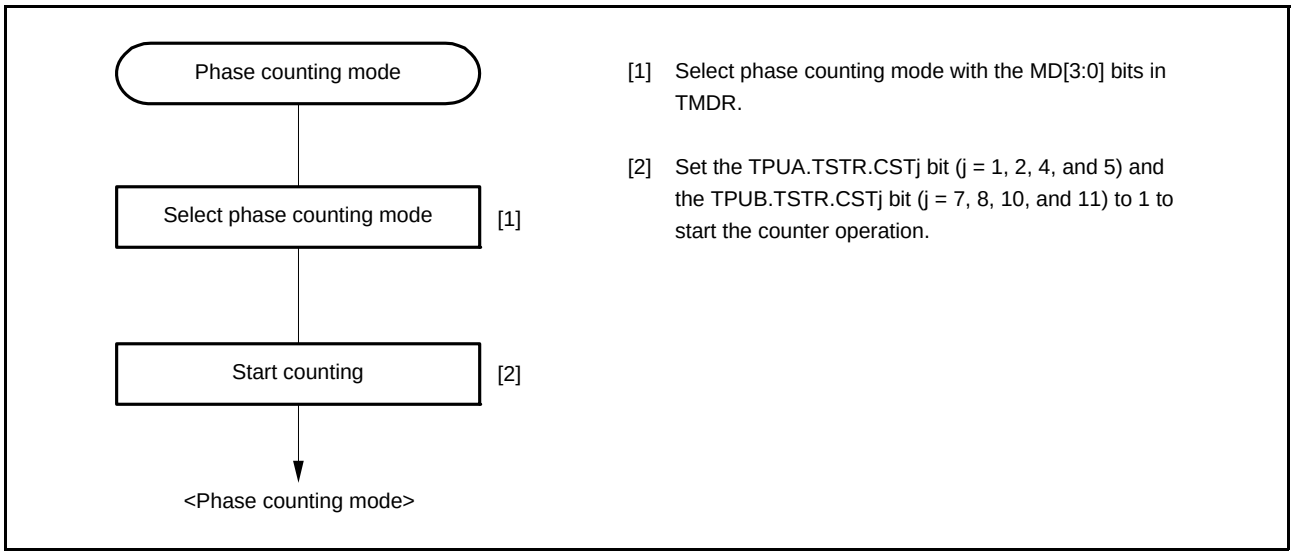


Figure 24.25 Example of Phase Counting Mode Setting Procedure

(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 24.26 shows an example of phase counting mode 1 operation, and Table 24.26 lists the TPUm.TCNT up-/down-count conditions.

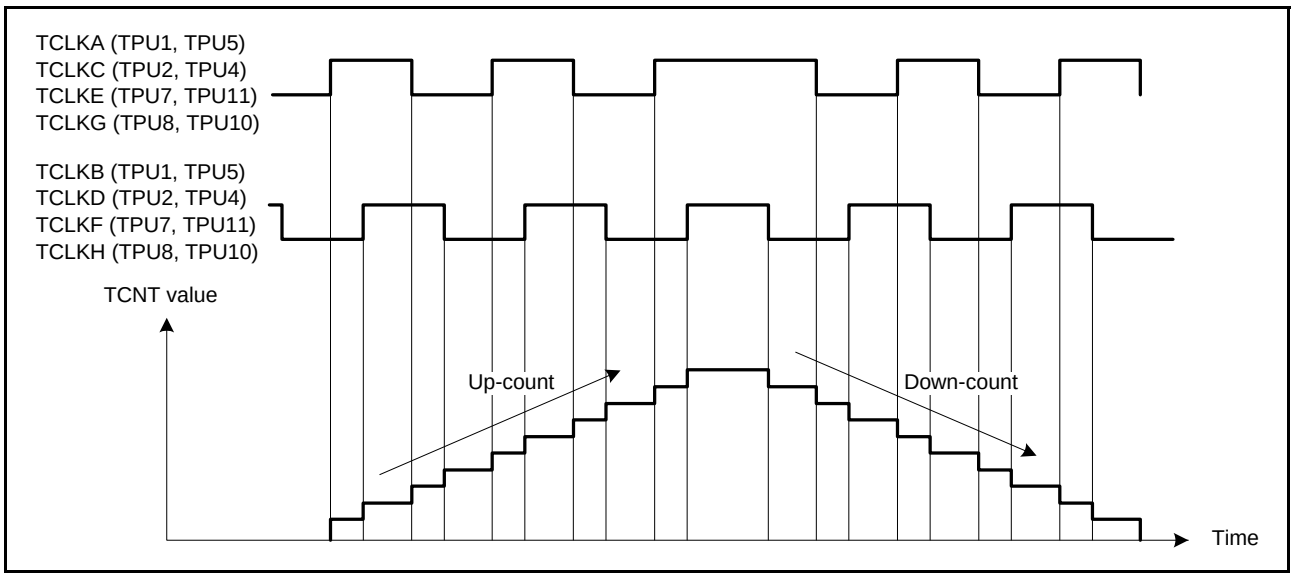


Figure 24.26 Example of Phase Counting Mode 1 Operation

Table 24.26 Up-/Down-Count Conditions in Phase Counting Mode 1

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4) TCLKE (TPU7, TPU11) TCLKG (TPU8, TPU10)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4) TCLKF (TPU7, TPU11) TCLKH (TPU8, TPU10)	Operation
High	↑	Up-count
Low	↓	
↑	Low	
↓	High	
High	↓	Down-count
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase counting mode 2

Figure 24.27 shows an example of phase counting mode 2 operation, and Table 24.27 lists the TPUm.TCNT up-/down-count conditions.

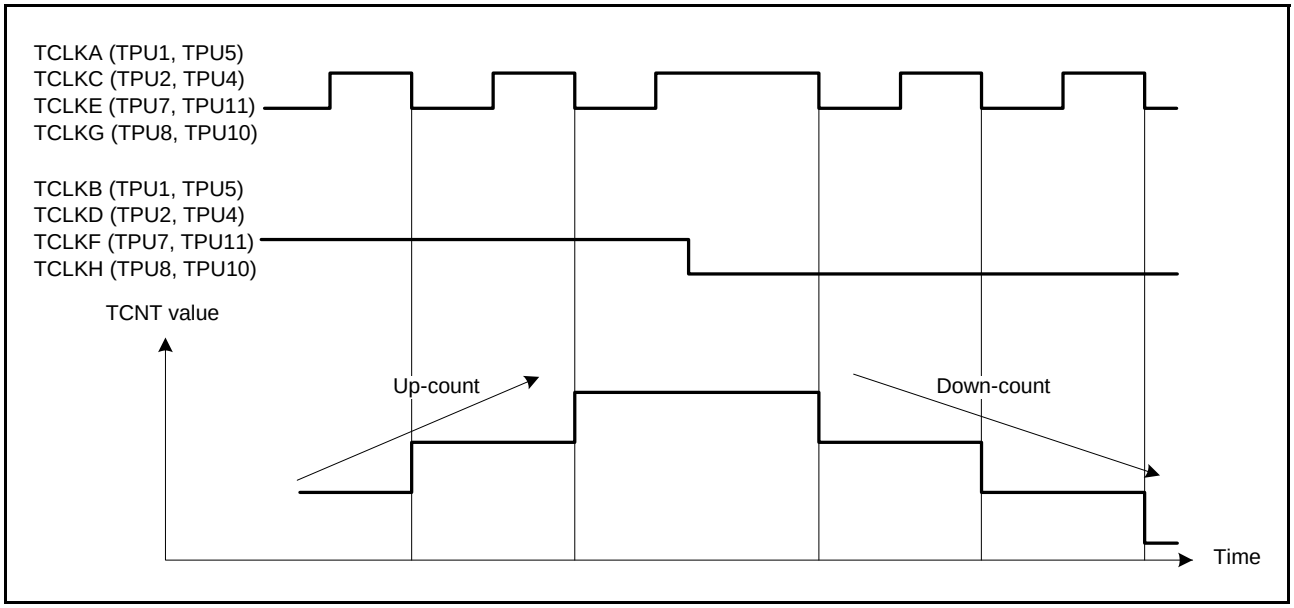


Figure 24.27 Example of Phase Counting Mode 2 Operation

Table 24.27 Up-/Down-Count Conditions in Phase Counting Mode 2

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4) TCLKE (TPU7, TPU11) TCLKG (TPU8, TPU10)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4) TCLKF (TPU7, TPU11) TCLKH (TPU8, TPU10)	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 24.28 shows an example of phase counting mode 3 operation, and Table 24.28 lists the TPUm.TCNT up-/down-count conditions.

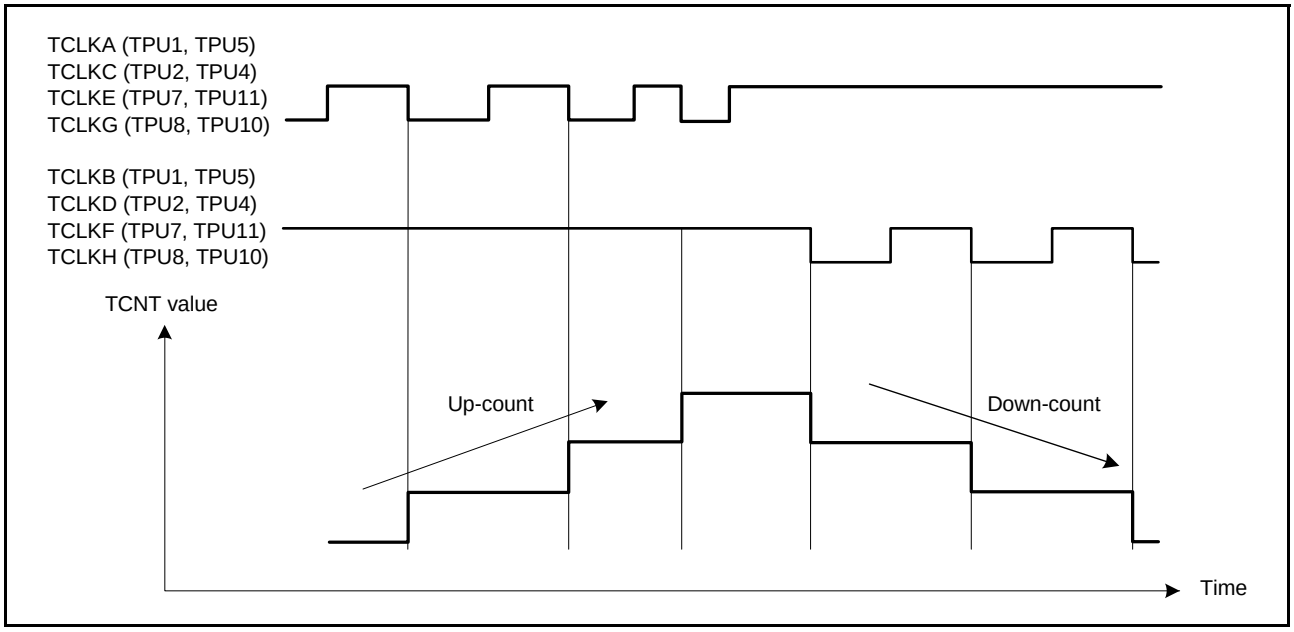


Figure 24.28 Example of Phase Counting Mode 3 Operation

Table 24.28 Up-/Down-Count Conditions in Phase Counting Mode 3

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4) TCLKE (TPU7, TPU11) TCLKG (TPU8, TPU10)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4) TCLKF (TPU7, TPU11) TCLKH (TPU8, TPU10)	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 24.29 shows an example of phase counting mode 4 operation, and Table 24.29 lists the TPUm.TCNT up-/down-count conditions.

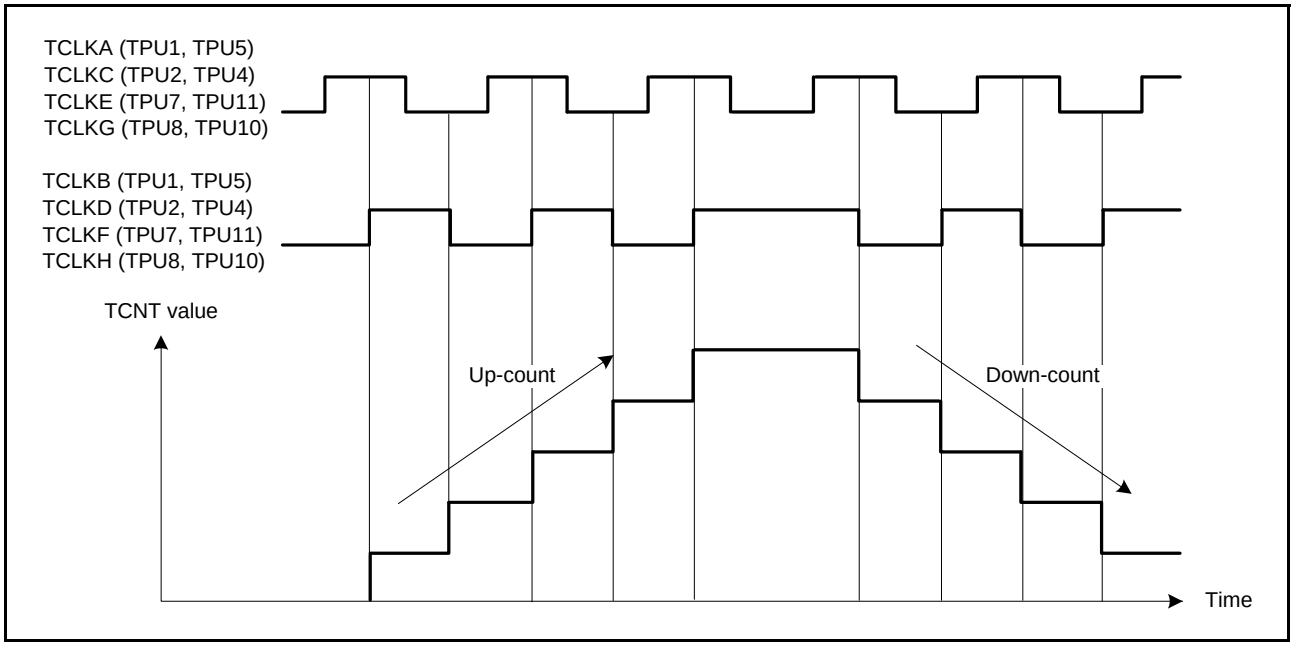


Figure 24.29 Example of Phase Counting Mode 4 Operation

Table 24.29 Up-/Down-Count Conditions in Phase Counting Mode 4

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4) TCLKE (TPU7, TPU11) TCLKG (TPU8, TPU10)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4) TCLKF (TPU7, TPU11) TCLKH (TPU8, TPU10)	Operation
High	↑	Up-count
Low	↓	
↑	Low	Don't care
↓	High	
High	↓	Down-count
Low	↑	
↑	High	Don't care
↓	Low	

↑ : Rising edge

↓ : Falling edge

24.3.6.1 Phase Counting Mode Application Example

Figure 24.30 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 counter 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 counter input 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.

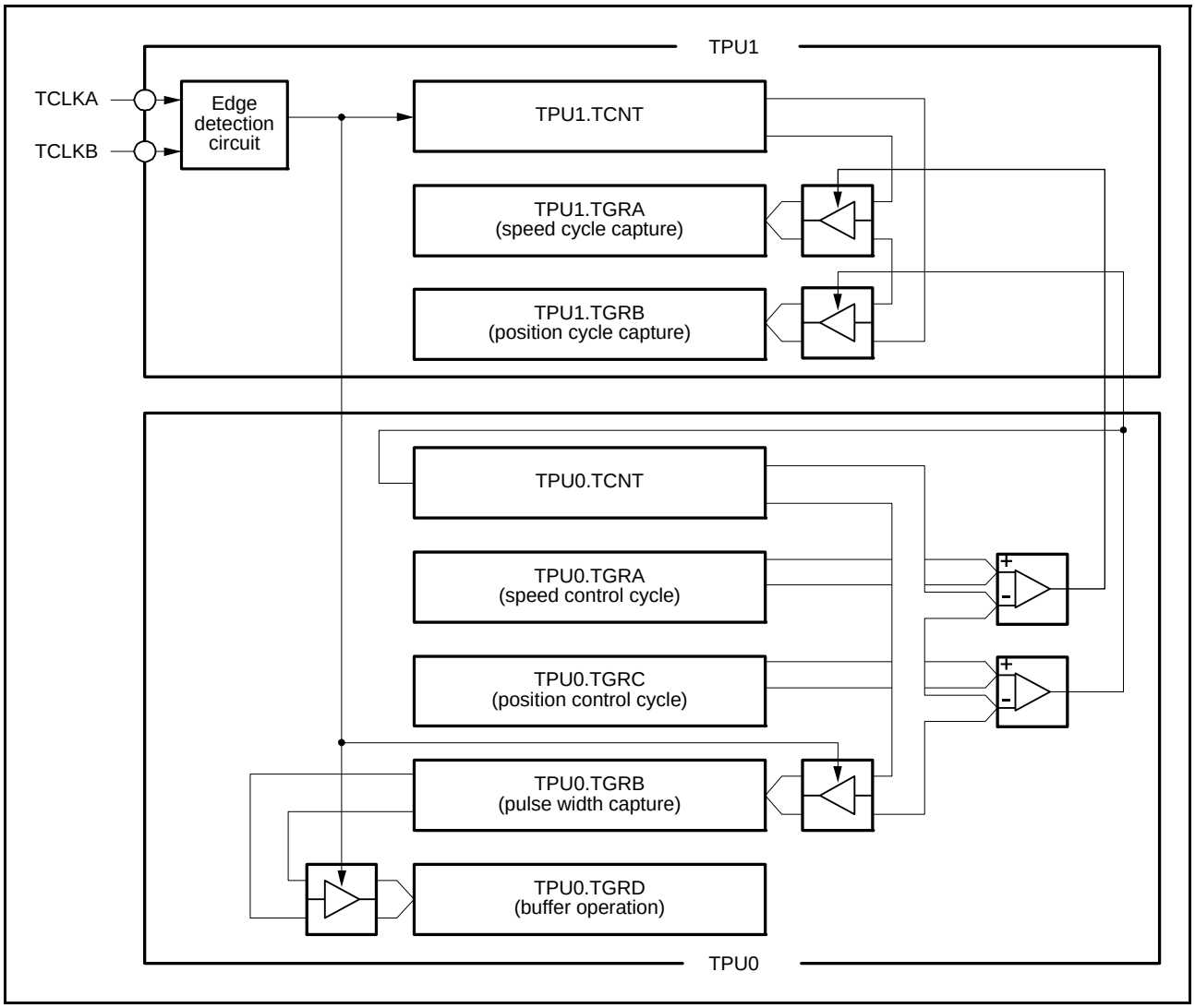


Figure 24.30 Phase Counting Mode Application Example

24.3.7 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 24.31 is a timing chart for the noise filter.

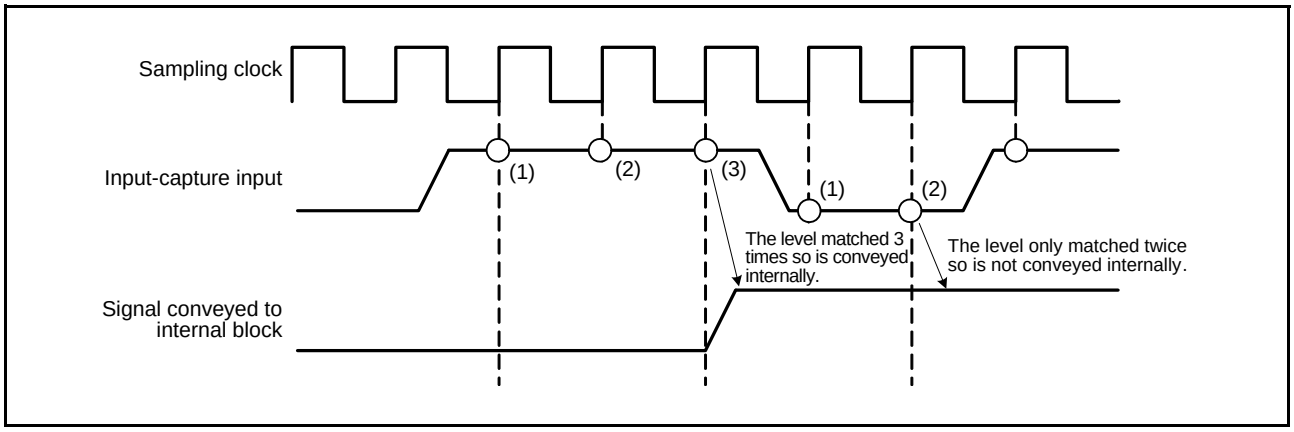


Figure 24.31 Timing Chart for the Noise Filter

24.4 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 (ICUb).

Table 24.30 lists the TPU interrupt sources.

Table 24.30 TPU Interrupt Sources (1/2)

Unit	Channel	Name	Interrupt Source	DTC Activation	DMAC Activation
0	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	TPU0A.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

Table 24.30 TPU Interrupt Sources (2/2)

Unit	Channel	Name	Interrupt Source	DTC Activation	DMAC Activation
1	TPU6	TGI6A	TPU6.TGRA input capture/compare match	Possible	Possible
		TGI6B	TPU6.TGRB input capture/compare match	Possible	Not possible
		TGI6C	TPU6.TGRC input capture/compare match	Possible	Not possible
		TGI6D	TPU6.TGRD input capture/compare match	Possible	Not possible
		TCI6V	TPU6.TCNT overflow	Not possible	Not possible
	TPU7	TGI7A	TPU7.TGRA input capture/compare match	Possible	Possible
		TGI7B	TPU7.TGRB input capture/compare match	Possible	Not possible
		TCI7V	TPU7.TCNT overflow	Not possible	Not possible
		TCI7U	TPU7.TCNT underflow	Not possible	Not possible
	TPU8	TGI8A	TPU8.TGRA input capture/compare match	Possible	Possible
		TGI8B	TPU8.TGRB input capture/compare match	Possible	Not possible
		TCI8V	TPU8.TCNT overflow	Not possible	Not possible
		TCI8U	TPU8.TCNT underflow	Not possible	Not possible
	TPU9	TGI9A	TPU9.TGRA input capture/compare match	Possible	Possible
		TGI9B	TPU9.TGRB input capture/compare match	Possible	Not possible
		TGI9C	TPU9.TGRC input capture/compare match	Possible	Not possible
		TGI9D	TPU9.TGRD input capture/compare match	Possible	Not possible
		TCI9V	TPU9.TCNT overflow	Not possible	Not possible
	TPU10	TGI10A	TPU10.TGRA input capture/compare match	Possible	Possible
		TGI10B	TPU10.TGRB input capture/compare match	Possible	Not possible
		TCI10V	TPU10.TCNT overflow	Not possible	Not possible
		TCI10U	TPU10.TCNT underflow	Not possible	Not possible
	TPU11	TGI11A	TPU11.TGRA input capture/compare match	Possible	Possible
		TGI11B	TPU11.TGRB input capture/compare match	Possible	Not possible
		TCI11V	TPU11.TCNT overflow	Not possible	Not possible
		TCI11U	TPU11.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 32 input capture/compare match interrupts, four each for TPU0, TPU3, TPU6 and TPU9, and two each for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11.

(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 twelve 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 eight underflow interrupts, one each for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11.

24.5 DTC Activation

The DTC can be activated by the TPUm.TGRy input capture/compare match interrupt of each channel. For details, see section 19, Data Transfer Controller (DTCa).
A total of 32 input capture/compare match interrupts can be used as DTC activation sources, four each for TPU0, TPU3, TPU6, and TPU9, and two each for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11.

24.6 DMAC Activation

The DMAC can be activated by the TPUm.TGRA input capture/compare match interrupt of each channel. For details, see section 17, DMA Controller (DMACA).
A total of twelve TPUm.TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

24.7 A/D Converter Activation

The TPU can activate the A/D converter by the TPUm.TGRA input capture/compare match for each channel. Moreover, the A/D converter is activated by the TGRA to TGRD input capture/compare match from TPU0 (TPU6).
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. Moreover, when the TGRA to TGRD input capture/compare match occurs in TPU0 (TPU6), the TPU requests the corresponding A/D converter to start A/D conversion. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started. These four A/D conversion start request signals cannot enable or disable the request signal generation by TIER.TTGE bit.
For the corresponding unit of A/D converter, see section 41, 10-Bit A/D Converter (ADb).

24.8 PPG Trigger

Input capture to or compare match with TGRA and TGRB in TPU0 to TPU3 (or TPUT6 to TPU8) can be made to act as a PPG1 waveform trigger.

24.9 Operation Timing

24.9.1 Input/Output Timing

(1) TPUm.TCNT Count Timing

Figure 24.32 shows TPUm.TCNT count timing in internal clock operation, and Figure 24.33 shows TCNT count timing in external clock operation.

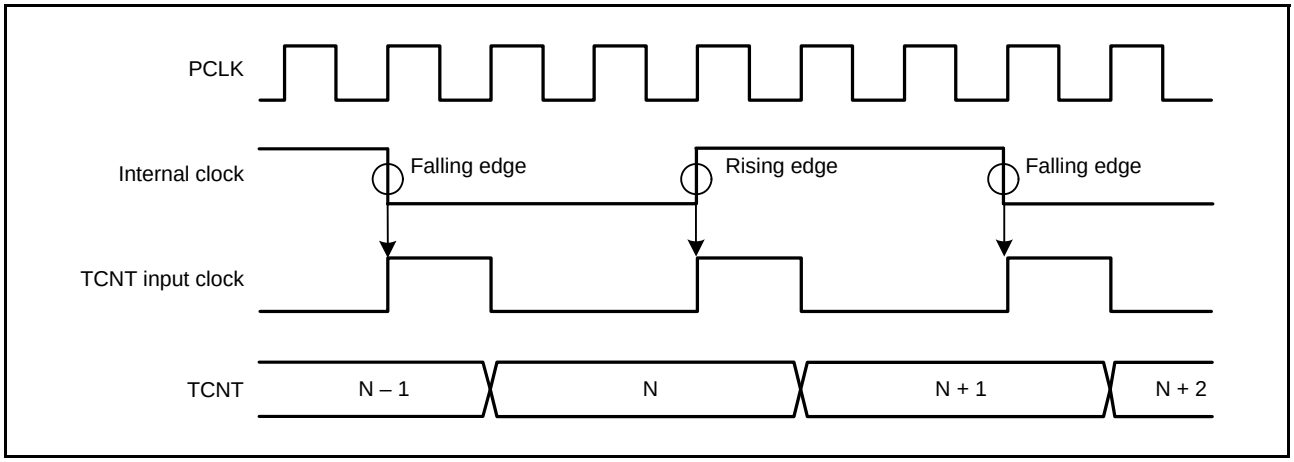


Figure 24.32 Count Timing in Internal Clock Operation

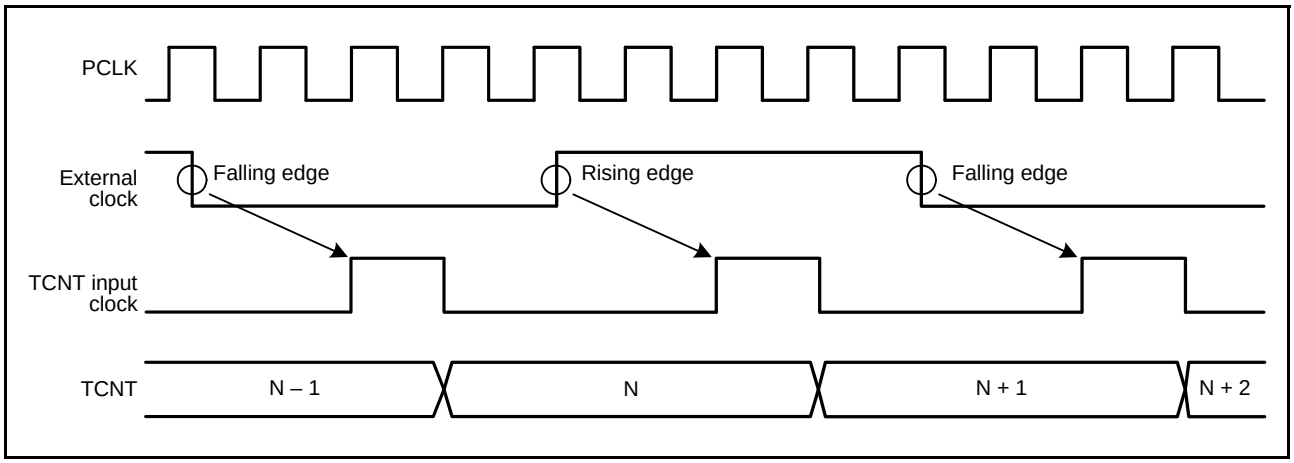


Figure 24.33 Count Timing in External Clock Operation

(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 11). After a match between TCNT and TGRy, the compare match signal is not generated until the TCNT input clock is generated.

Figure 24.34 shows output compare output timing.

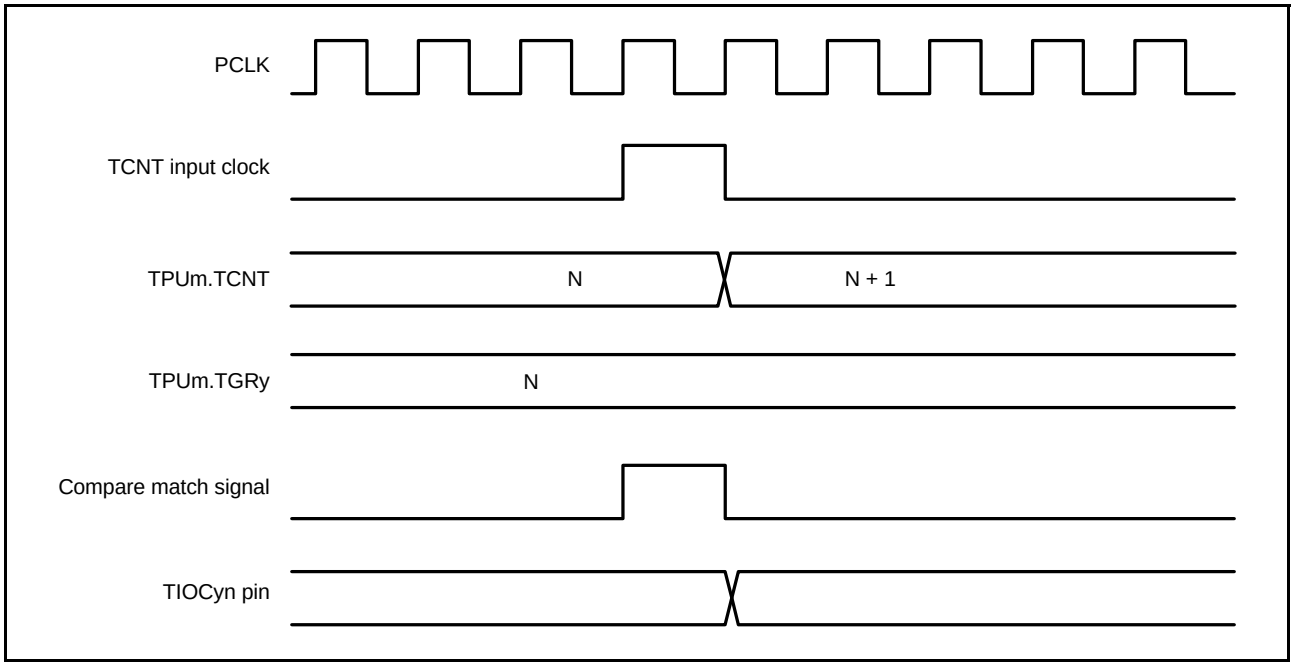


Figure 24.34 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 24.35 shows input capture signal timing.

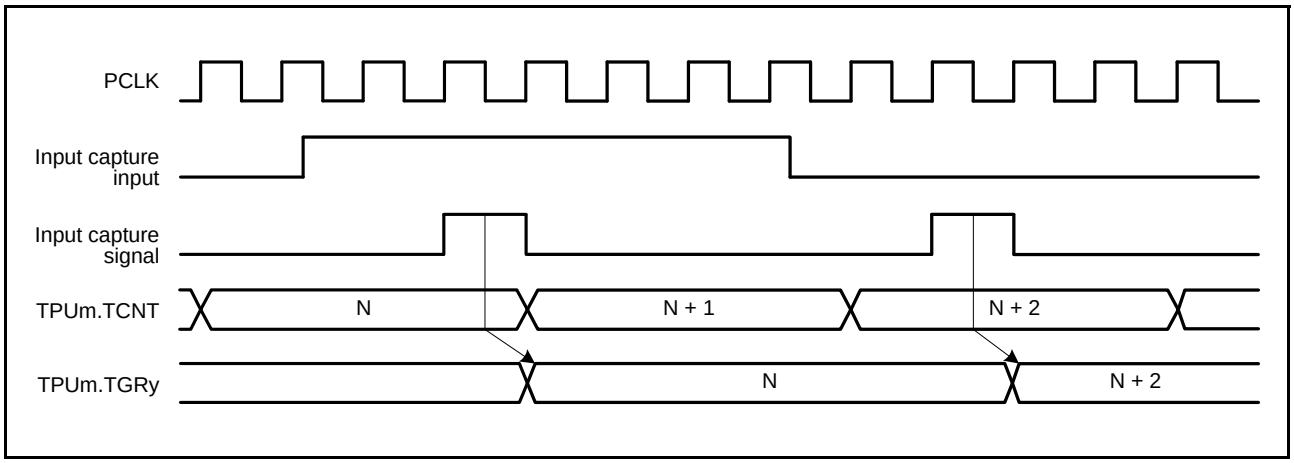


Figure 24.35 Input Capture Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 24.36 shows the timing when counter clearing by compare match occurrence is specified, and Figure 24.37 shows the timing when counter clearing by input capture occurrence is specified.

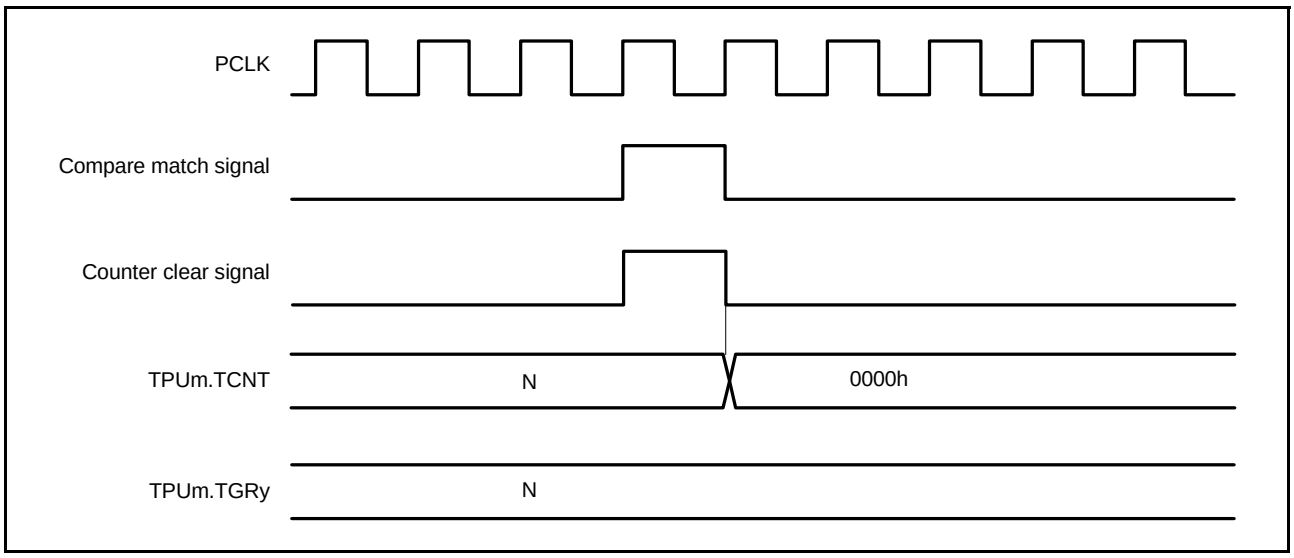


Figure 24.36 Counter Clear Timing (Compare Match)

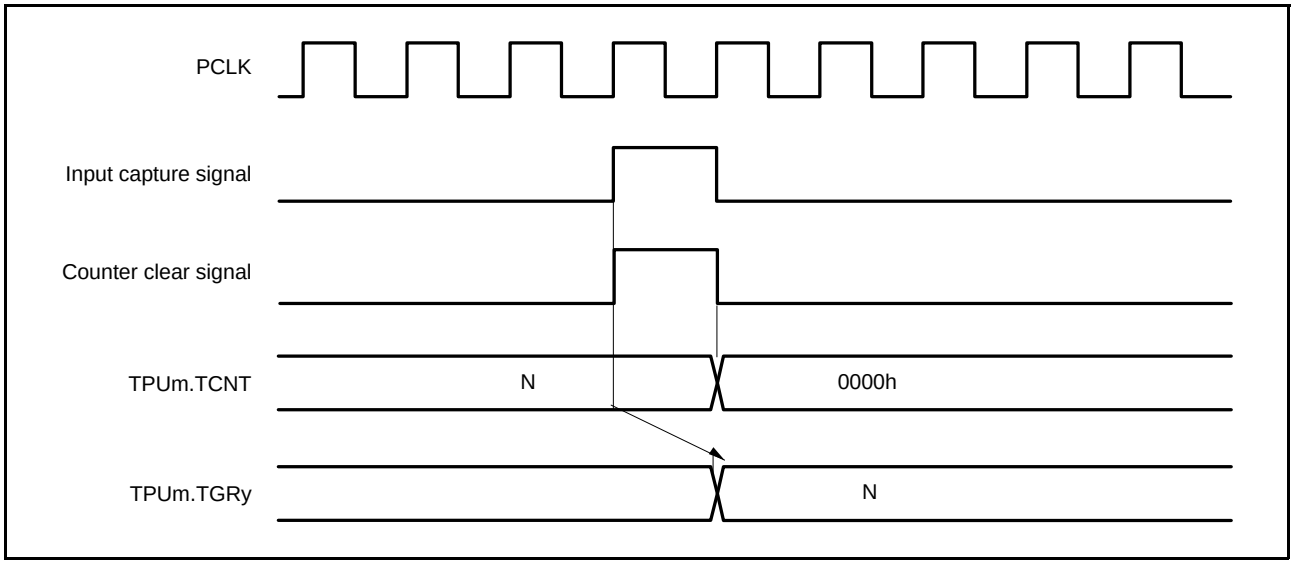


Figure 24.37 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figure 24.38 and Figure 24.39 show the timings in buffer operation.

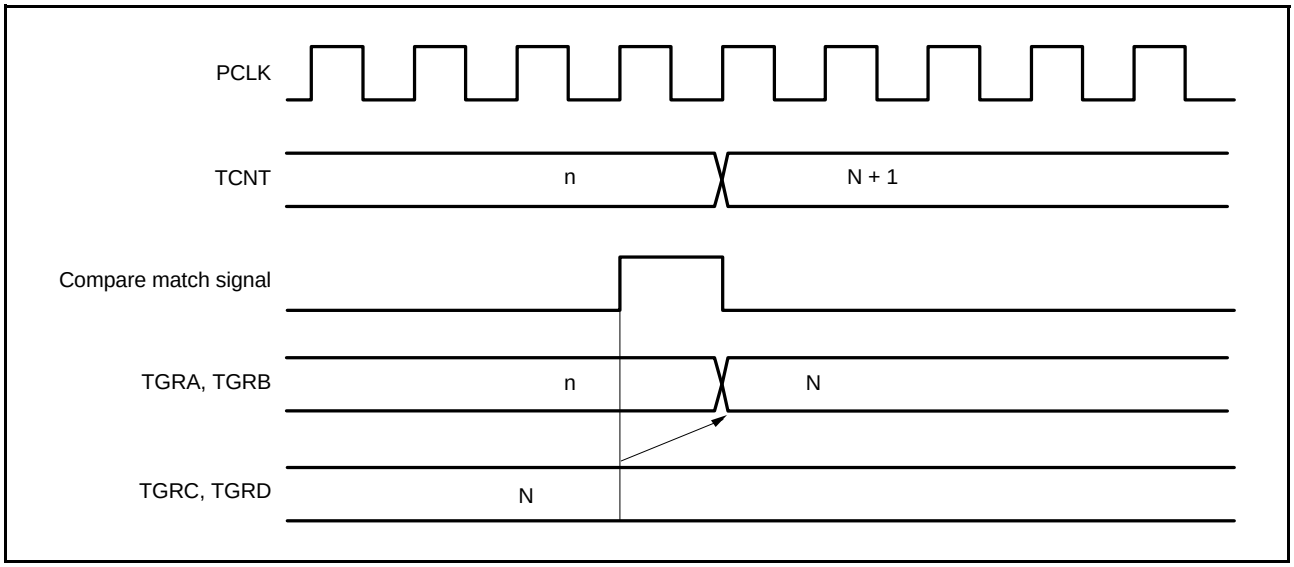


Figure 24.38 Buffer Operation Timing (Compare Match)

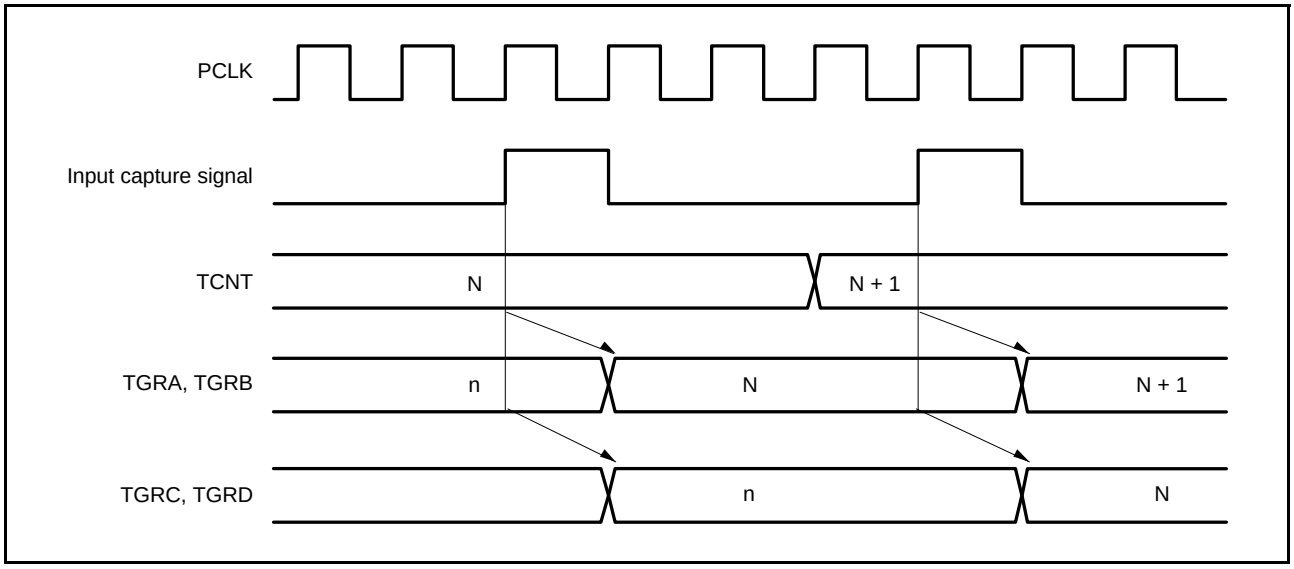


Figure 24.39 Buffer Operation Timing (Input Capture)

24.9.2 Interrupt Signal Timing

(1) Interrupt Flag Setting to 1 in Case of Compare Match

Figure 24.40 shows the timing for setting the interrupt flag by compare match occurrence.

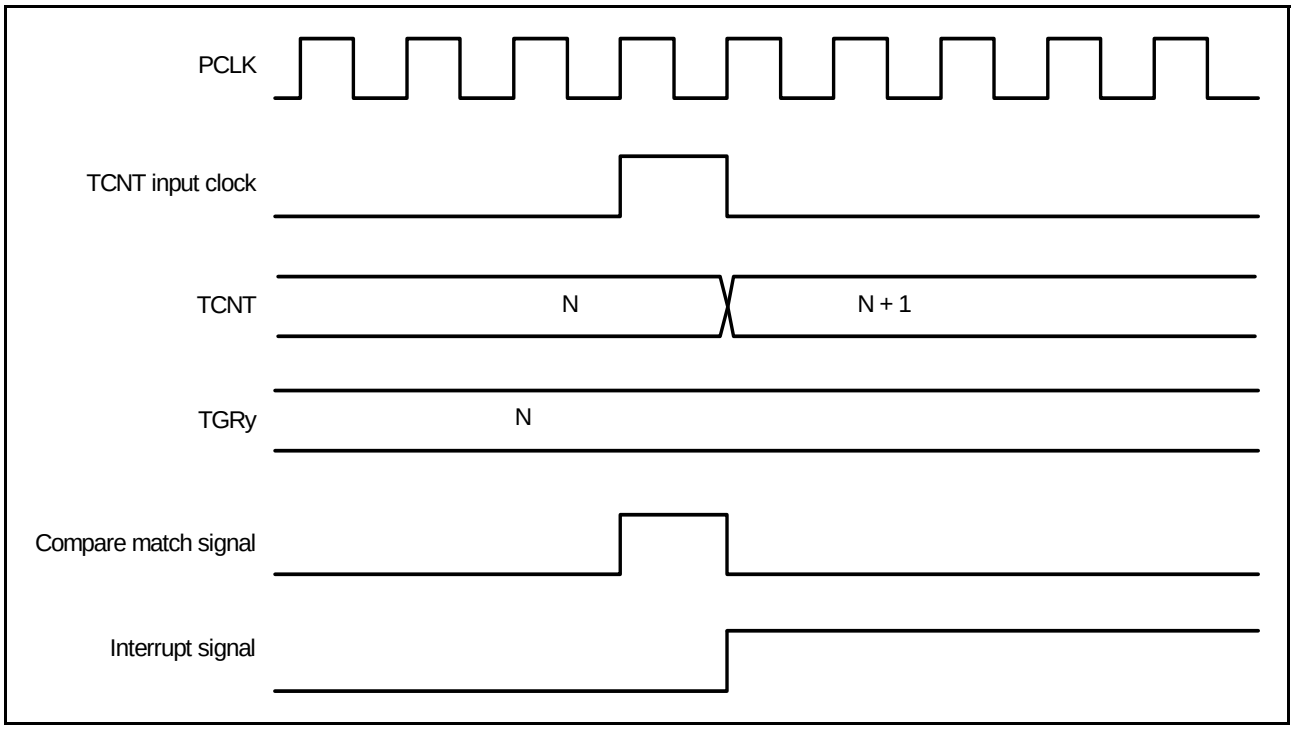


Figure 24.40 TGImy Interrupt Timing (Compare Match)

(2) Interrupt Flag Setting to 1 in Case of Input Capture

Figure 24.41 shows the timing for setting the interrupt flag by input capture occurrence.

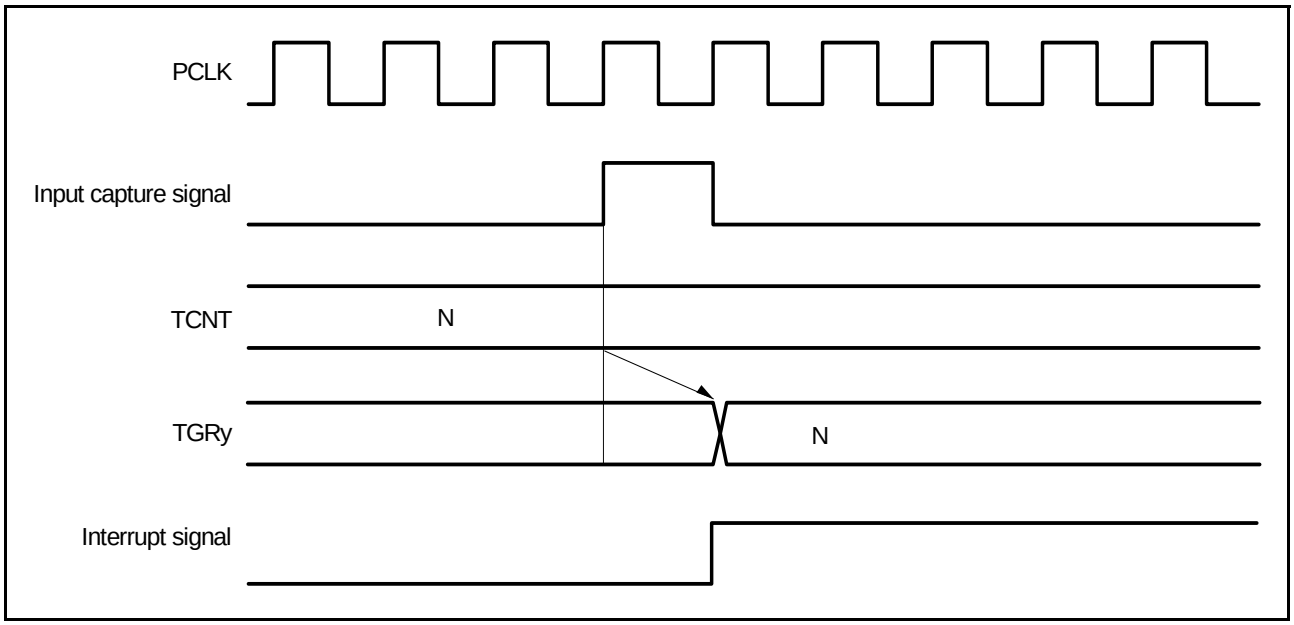


Figure 24.41 TGImy Interrupt Timing (Input Capture)

(3) TCImV/TCImU Interrupt Flag Setting to 1

Figure 24.42 shows the timing for generating the TCImV interrupt request signal by overflow occurrence.
Figure 24.43 shows the timing for generating the TCImU interrupt request signal by underflow occurrence.

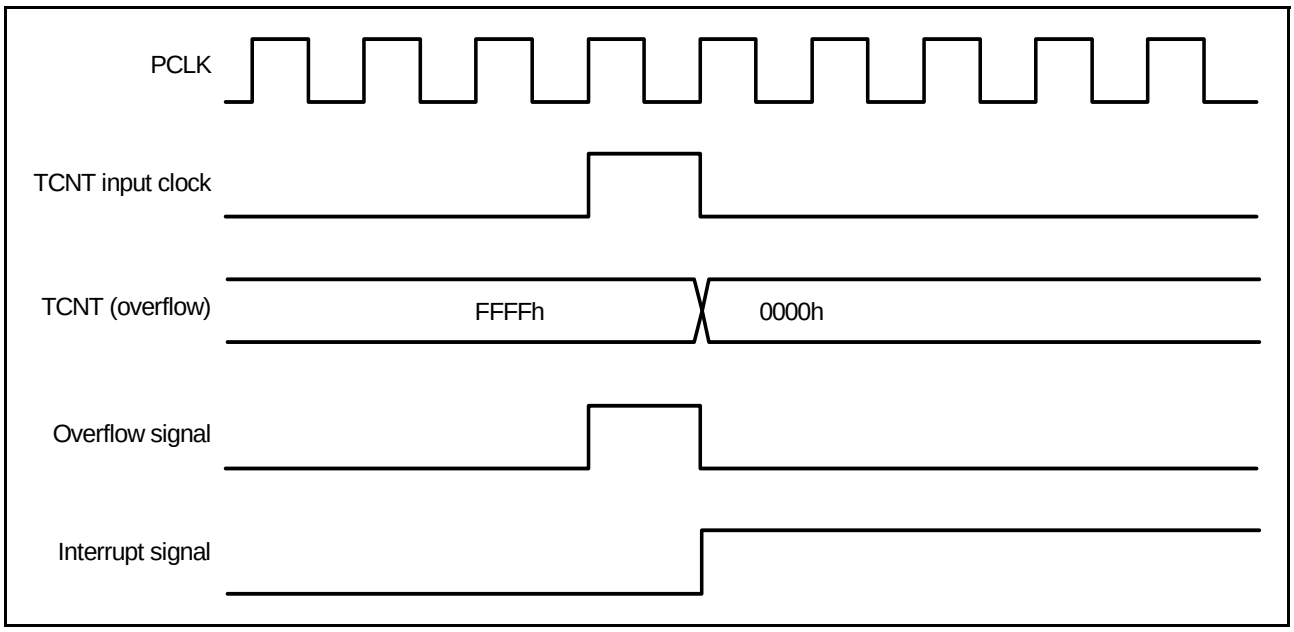


Figure 24.42 TCImV Interrupt Setting Timing

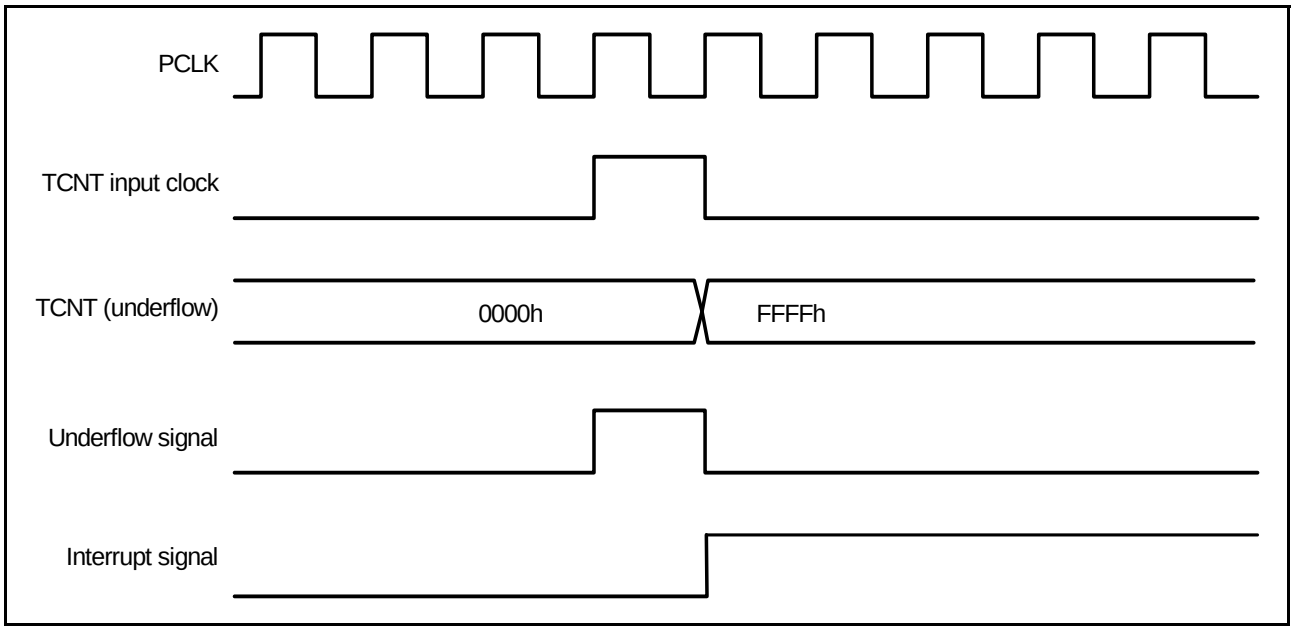


Figure 24.43 TCImU Interrupt Setting Timing

24.10 Usage Notes

24.10.1 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 clearing module stop state. For details, see section 11, Low Power Consumption.

24.10.2 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 TPUA 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 24.44 shows the input clock conditions in phase counting mode.

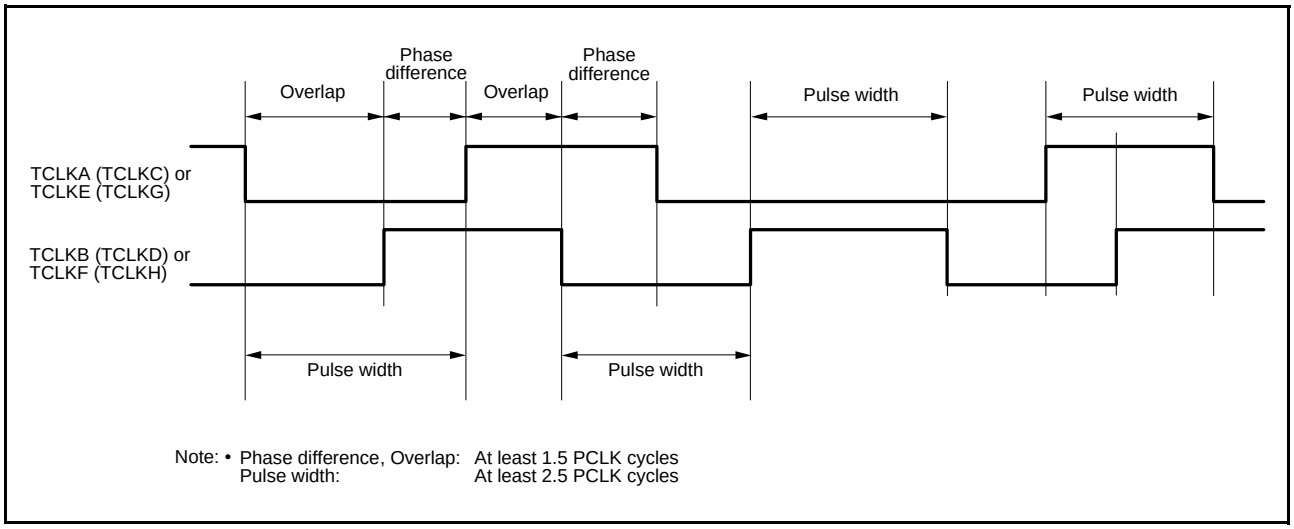


Figure 24.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

24.10.3 Caution 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:

$$f = \frac{PCLK}{(N+1)}$$

f: Counter frequency
PCLK: Operating frequency
N: TGRy set value

24.10.4 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 24.45 shows the timing in this case.

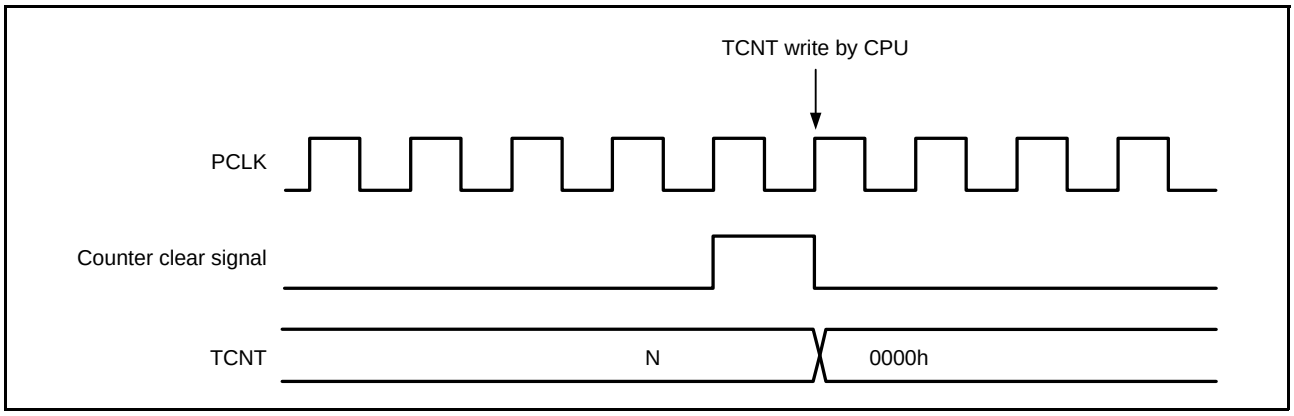


Figure 24.45 Conflict between TPUm.TCNT Write and Clear Operations

24.10.5 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 24.46 shows the timing in this case.

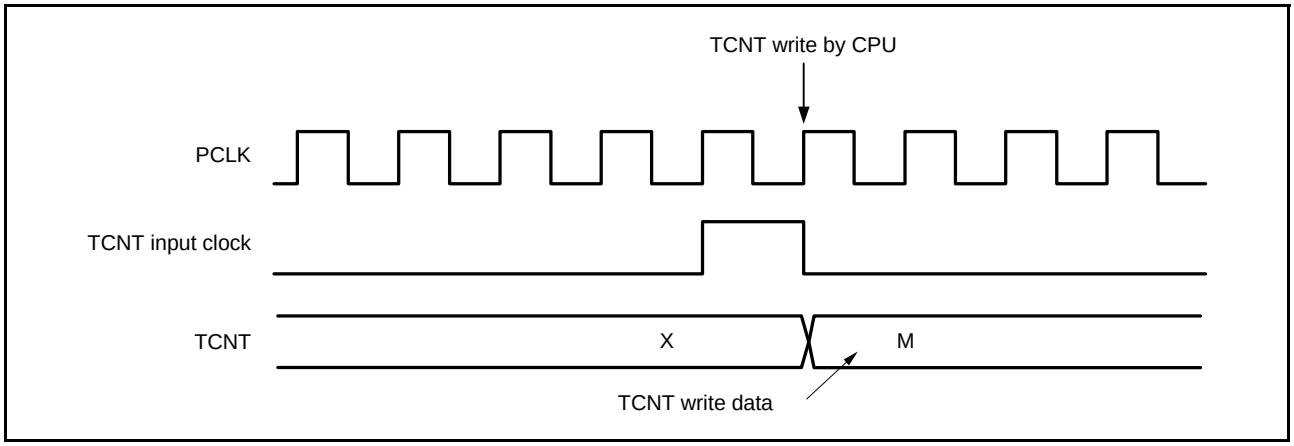


Figure 24.46 Conflict between TPUm.TCNT Write and Increment Operations

24.10.6 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 24.47 shows the timing in this case.

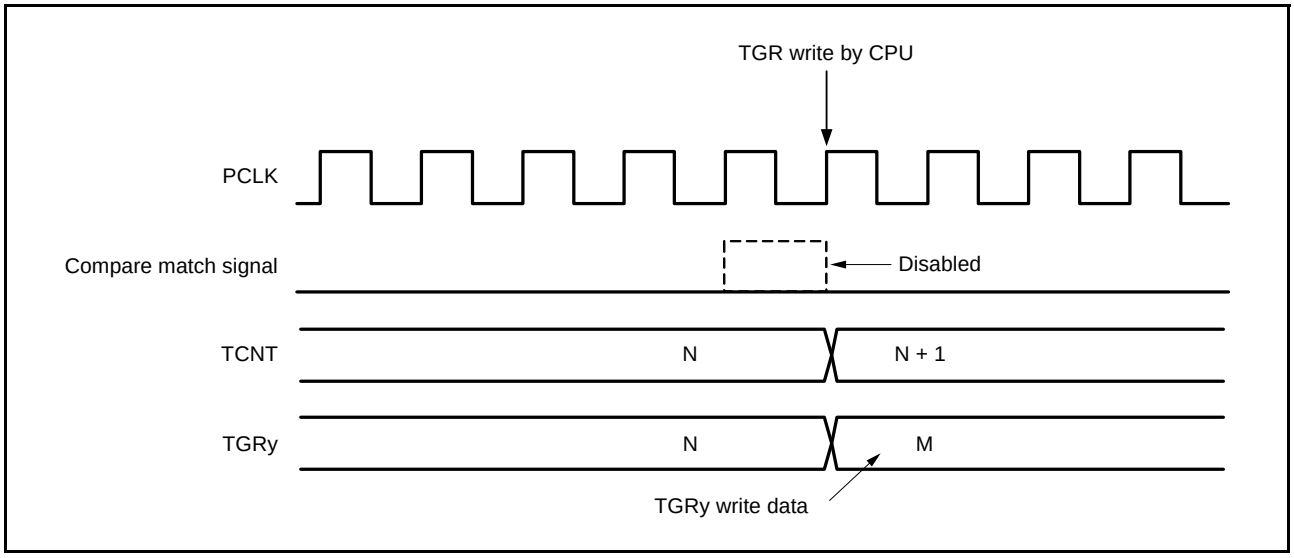


Figure 24.47 Conflict between TPUm.TGRy Write and Compare Match

24.10.7 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 24.48 shows the timing in this case.

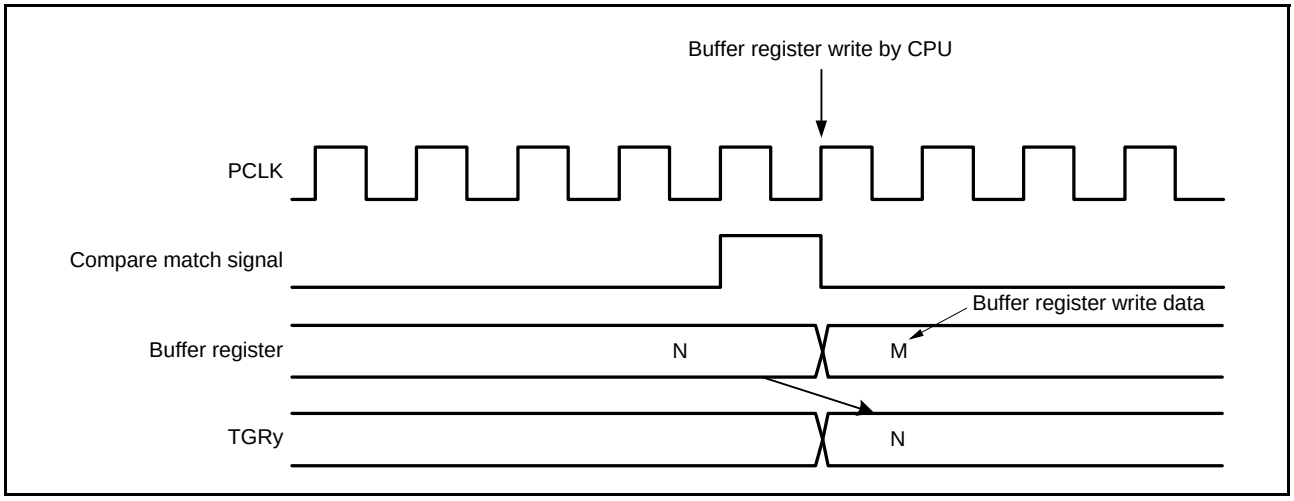


Figure 24.48 Conflict between Buffer Register Write and Compare Match

24.10.8 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 24.49 shows the timing in this case.

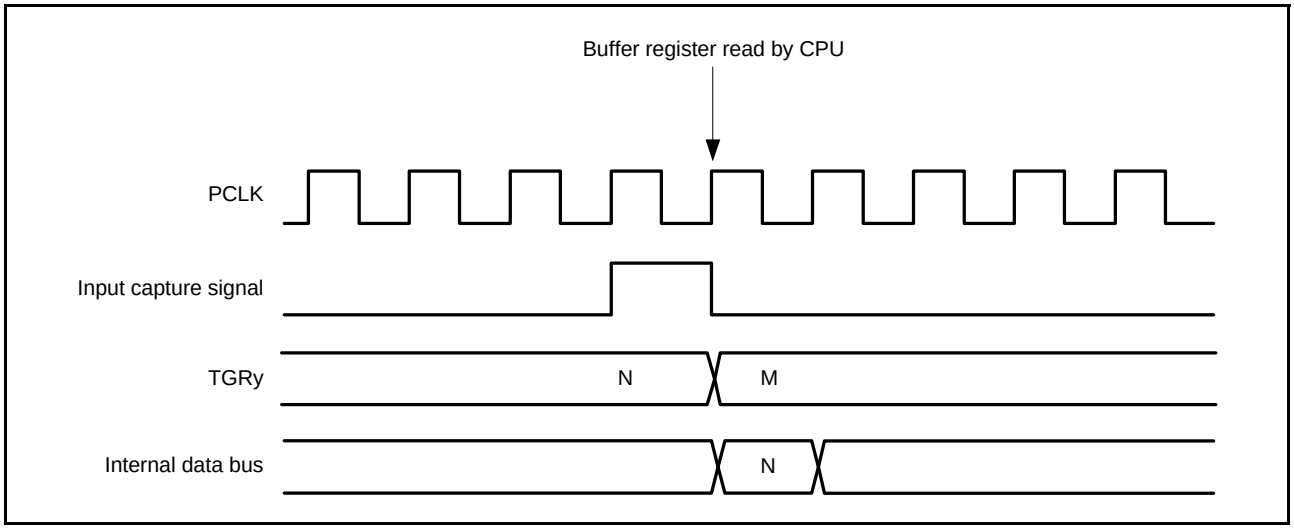


Figure 24.49 Conflict between TPUm.TGRy Read and Input Capture

24.10.9 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 24.50 shows the timing in this case.

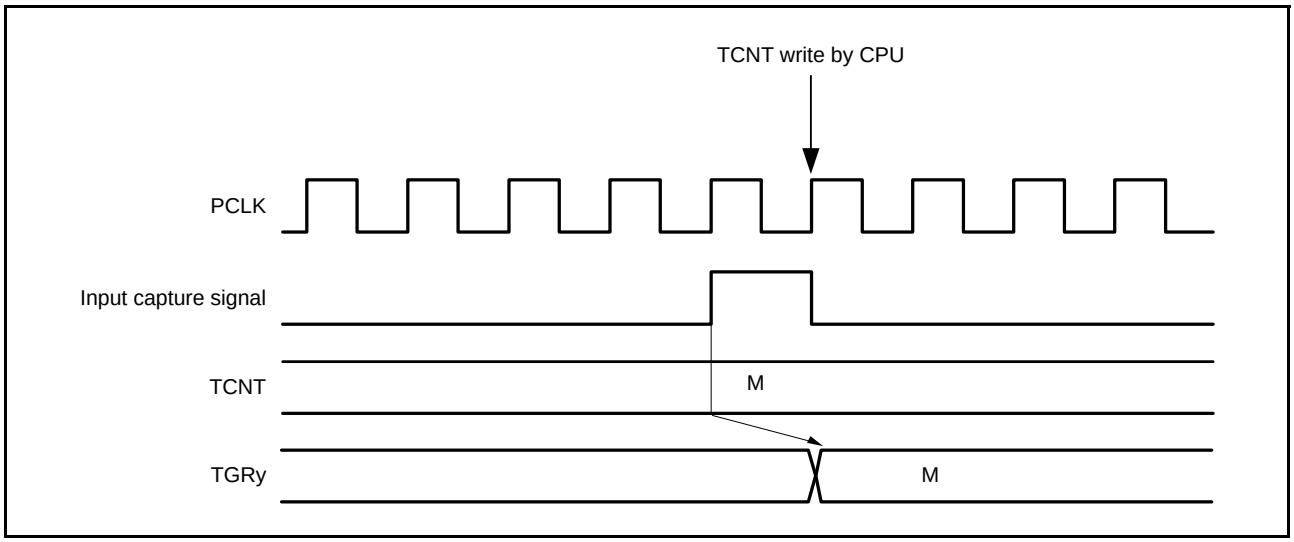


Figure 24.50 Conflict between TPUm.TGRy Write and Input Capture

24.10.10 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 24.51 shows the timing in this case.

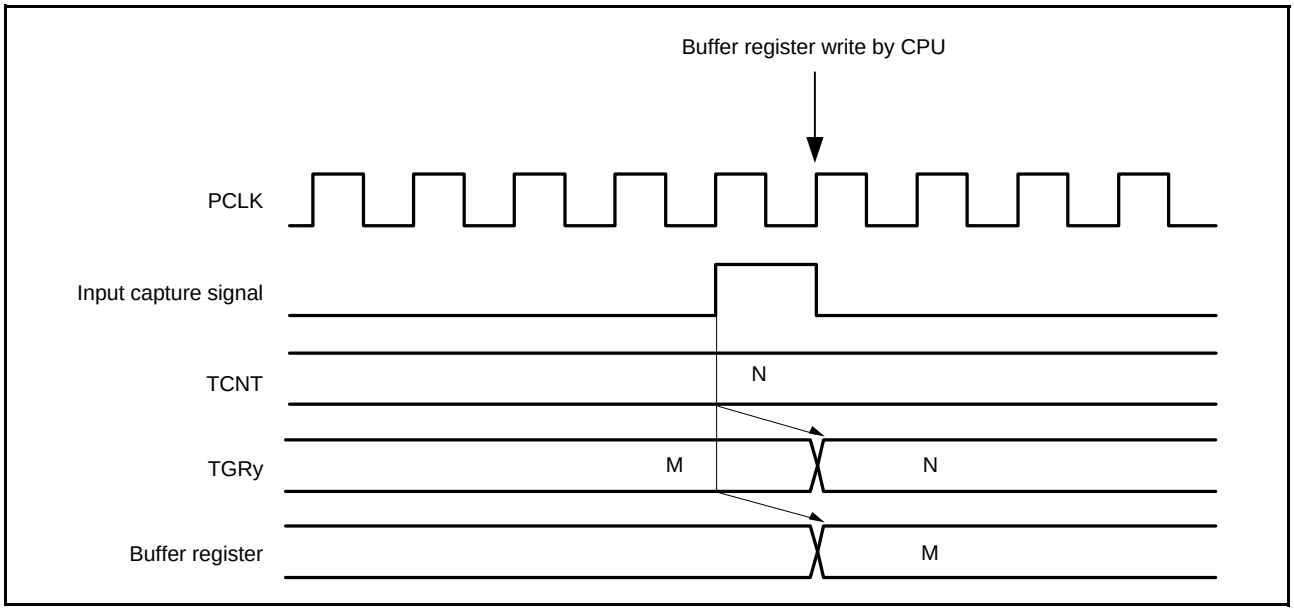
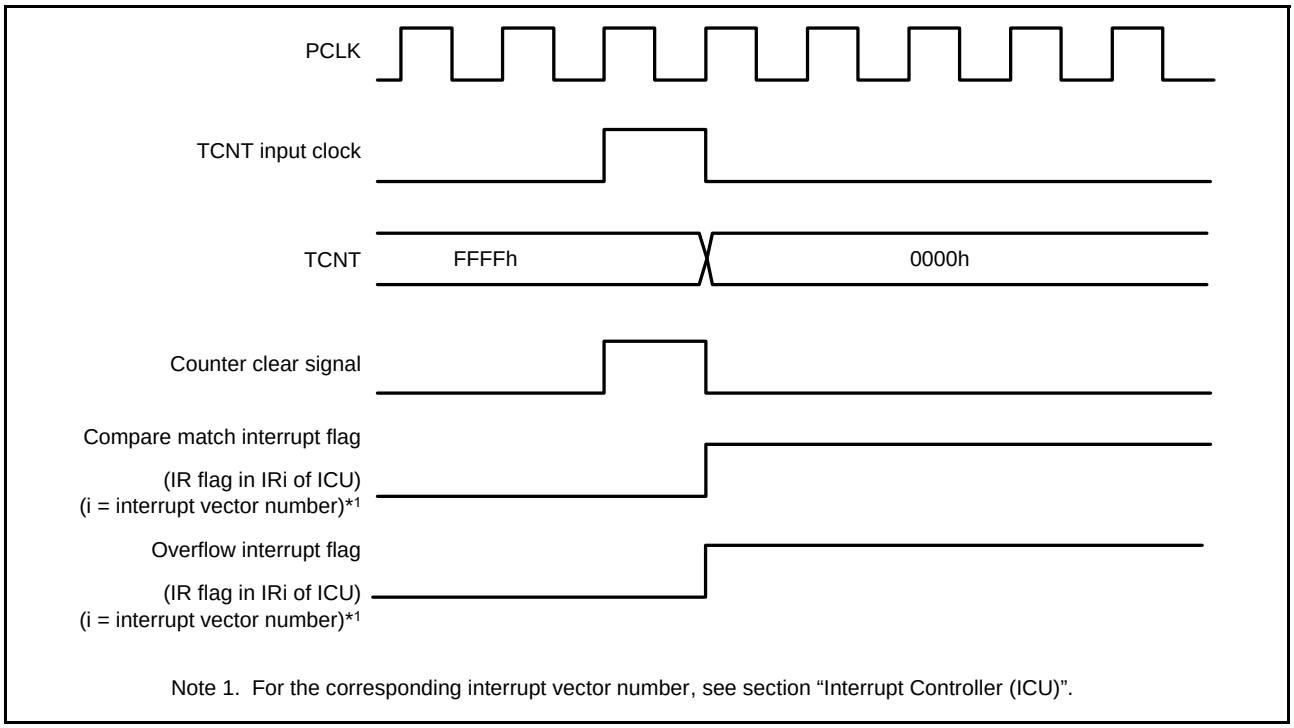


Figure 24.51 Conflict between Buffer Register Write and Input Capture

24.10.11 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, TPUm.TCNT counter is cleared with the generation of the compare match interrupt and an overflow interrupt is generated.
Figure 24.52 shows the operation timing when a TPUm.TGRy compare match is specified as the clearing source and FFFFh is set in TGRy.



Note 1. For the corresponding interrupt vector number, see section “Interrupt Controller (ICU)”.

Figure 24.52 Conflict between Overflow and Counter Clearing

24.10.12 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 24.53 shows the operation timing when there is conflict between TCNT write and overflow.

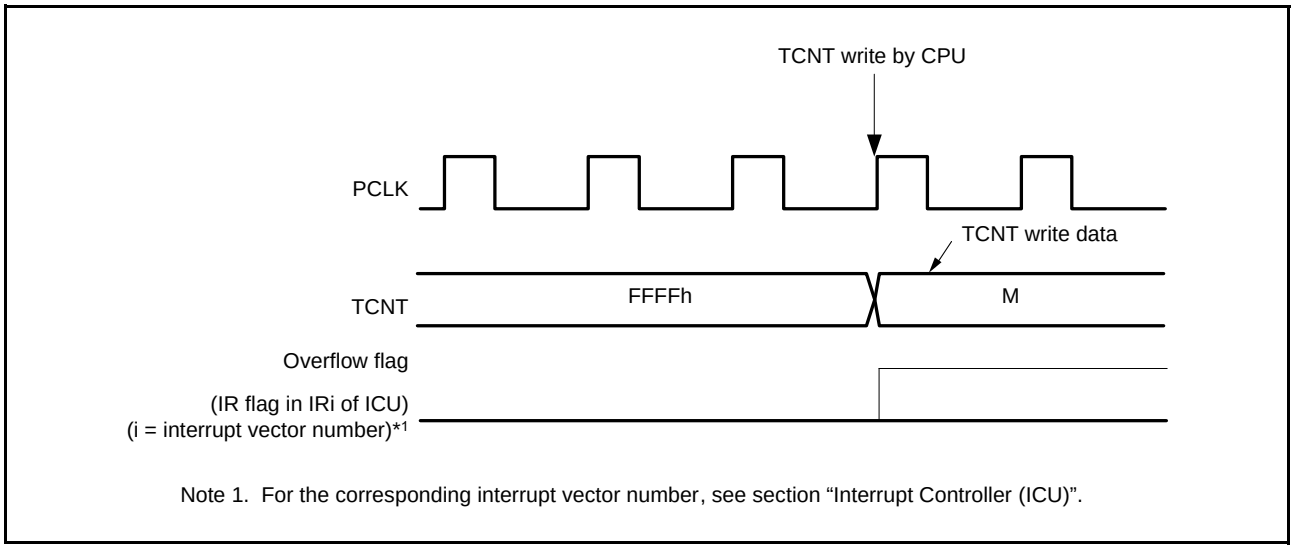


Figure 24.53 Conflict between TPUm.TCNT Write and Overflow

24.10.13 Multiplexing of I/O Pins

In the RX63N/RX631 Group, 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 TCLKG input pin with the TIOCB7 I/O pin, the TCLKH input pin with the TIOCB8 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCC3 I/O pin, the TCLKD input pin with the TIOCD3 I/O pin, the TCLKE input pin with the TIOCC6 I/O pin, and the TCLKF input pin with the TIOCD6 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

24.10.14 Continuous Output of Compare-Match Pulse Interrupt Signal

When TGR is set to 0, PCLK/1 is set as the counter clock, and compare match is set as the counter clear source, the TCNT counter 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 24.54 shows an operation timing when the compare-match pulse interrupt signal is continuously output.

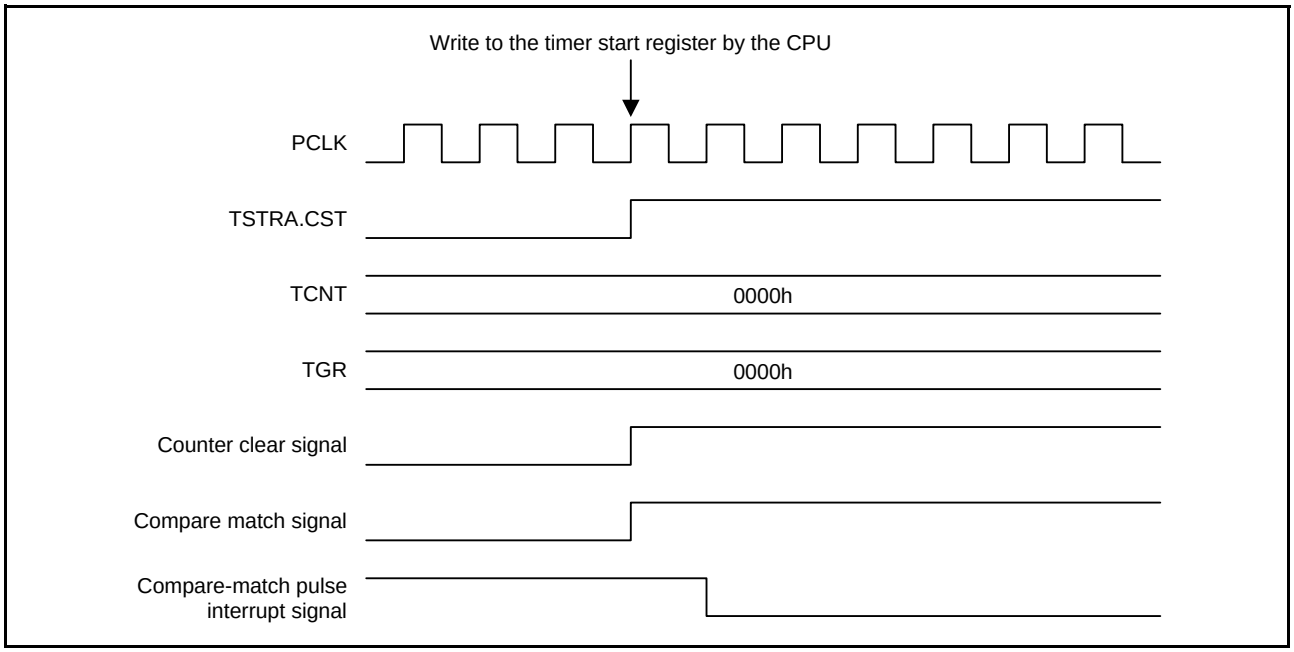


Figure 24.54 Continuous Output of Compare-Match Pulse Interrupt Signal

24.10.15 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 24.55 shows an operation timing when the input-capture pulse interrupt signal is output continuously.

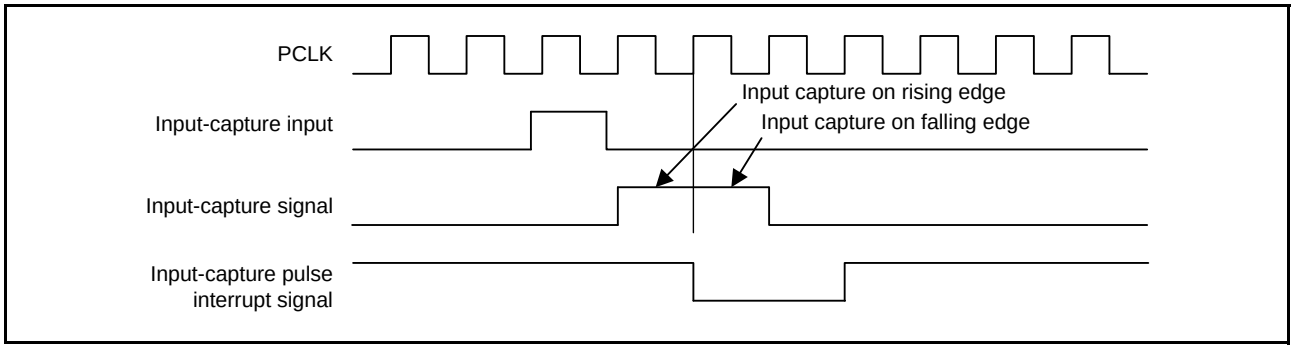


Figure 24.55 Continuous Output of Input-Capture Pulse Interrupt Signal

24.10.16 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 counter 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 24.56 shows an operation timing when the underflow pulse interrupt signal is output continuously.

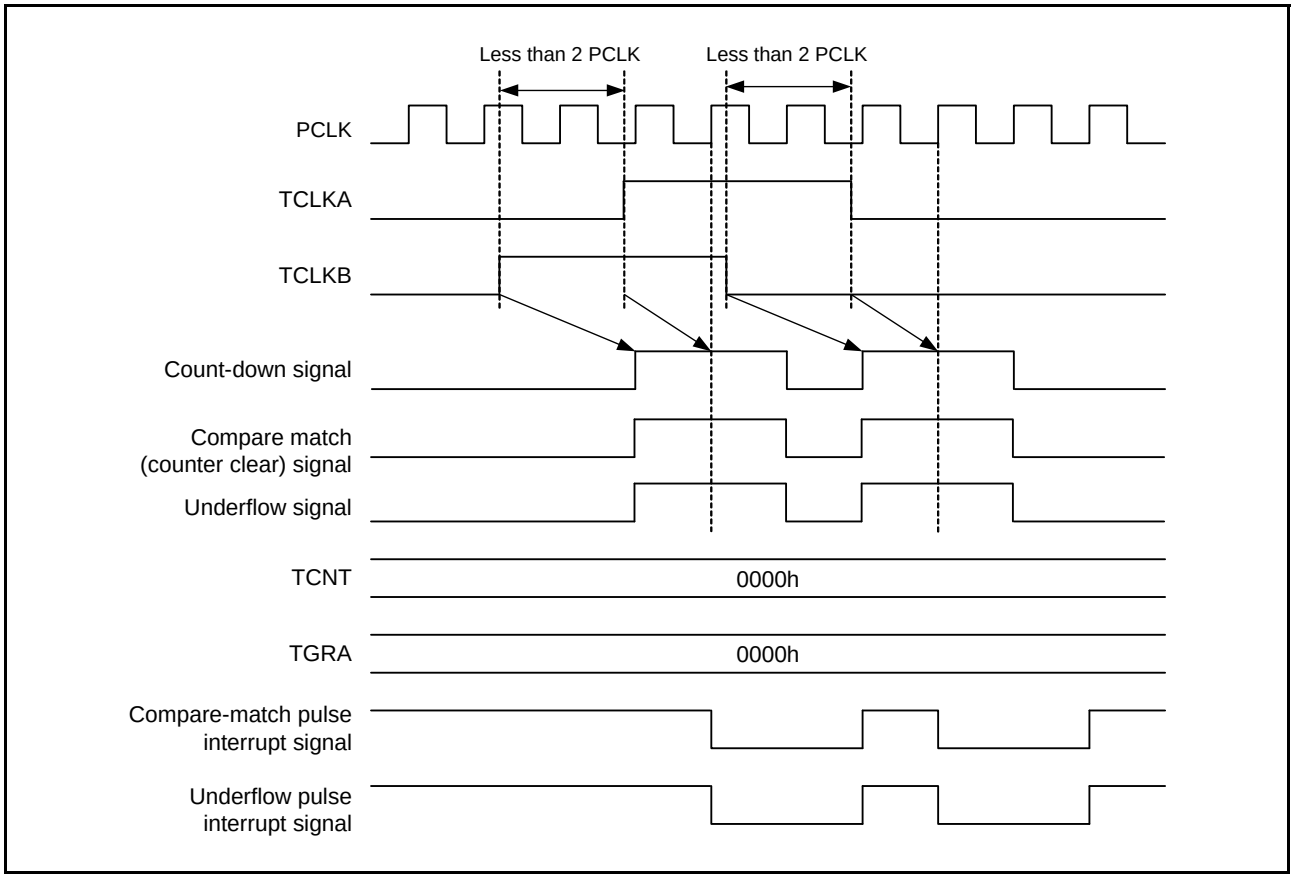


Figure 24.56 Continuous Output of Underflow Pulse Interrupt Signal

25. 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 2 (MTU) as a timebase.
The RX63N/RX631 Group 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.

25.1 Overview

Table 25.1 lists the specifications of the PPG and Table 25.2 lists PPG functions.
Figure 25.1 and Figure 25.2 show block diagrams of the PPGs.

Table 25.1 Specifications of PPG

Item	Specifications
Number of output bits	Up to 32 bits
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 MTU interrupts are in use)
Power consumption reducing function	Module stop state can be set for each unit.

Table 25.2 List of PPG Functions

Item			PPG0	PPG1
PPG output trigger	MTU channels 0 to 3 (MTU0 to MTU3)	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*1			The MSTPA11 bit in MSTPCRA	The MSTPA10 bit in MSTPCRA

○: Possible
—: Not possible
Note 1. For details, see section 11, Low Power Consumption.

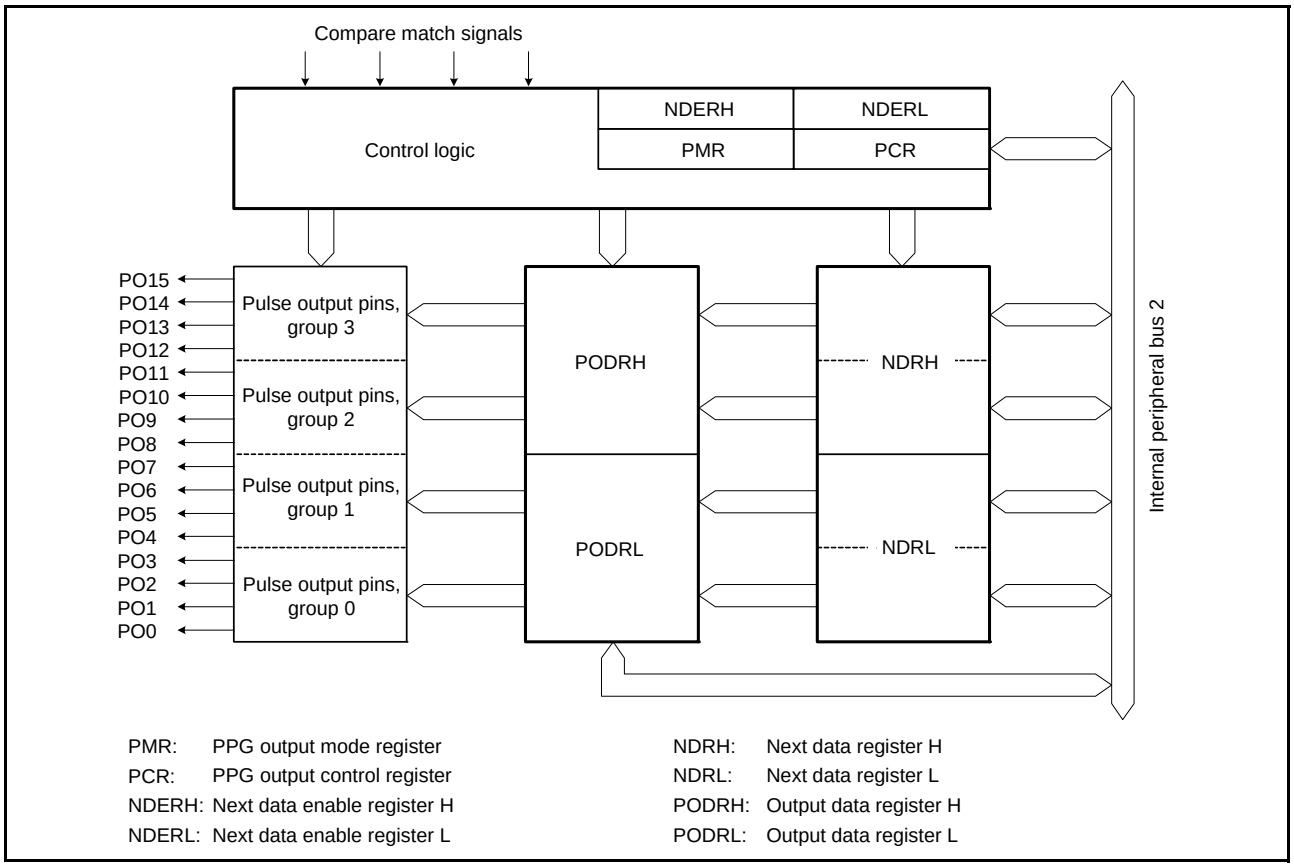


Figure 25.1 Block Diagram of PPG0

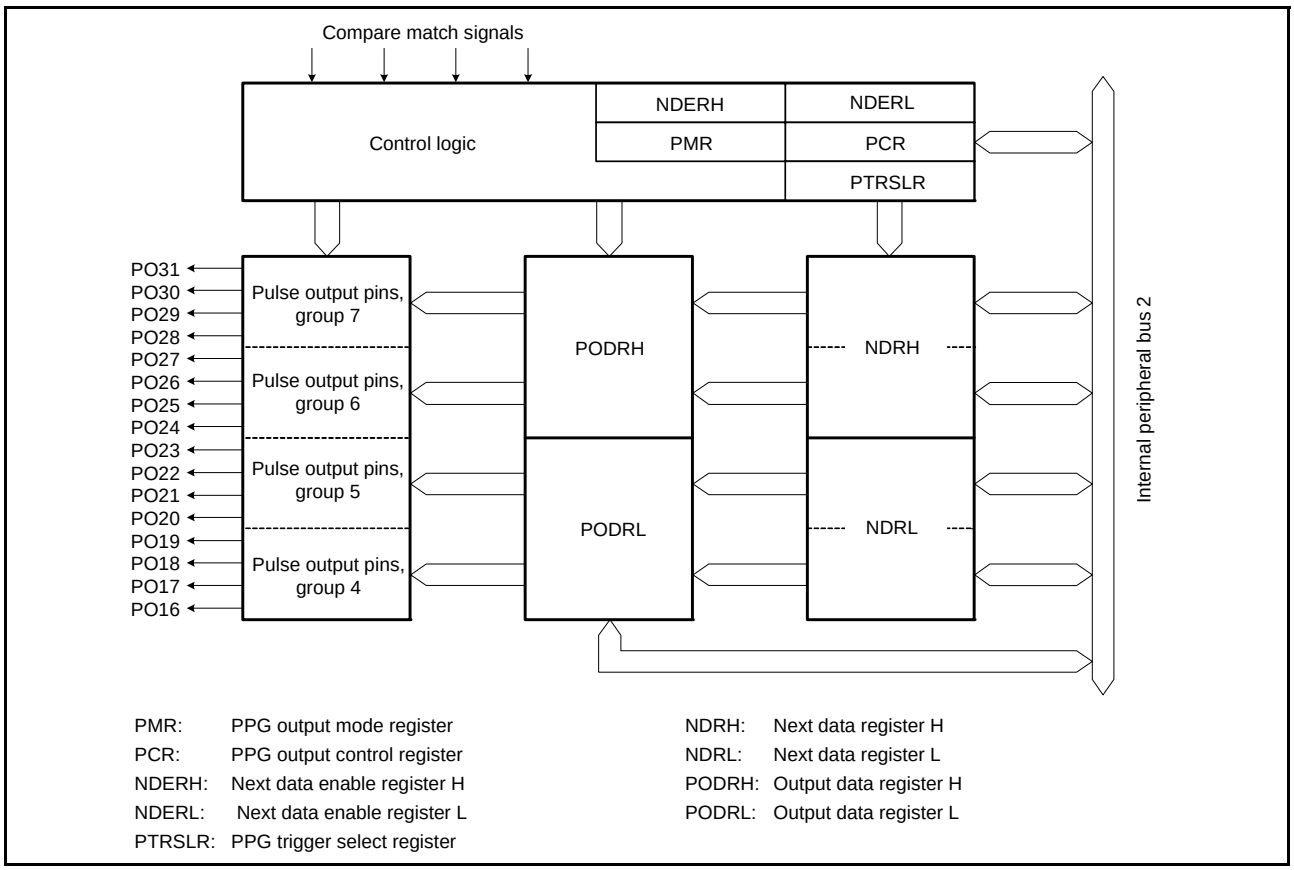


Figure 25.2 Block Diagram of PPG1

Table 25.3 lists the pin configuration of the PPG.

Table 25.3 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	

25.2 Register Descriptions

25.2.1 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 as the trigger channels for PPG1. 1: Selects the set of TPU0 to TPU3 as the trigger channels for PPG1.	R/W
b1 to 7	—	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 or TPU0 to TPU3 as a set of trigger channels for PPG1.
When this bit is set to 0, MTU0 to MTU3 are selected as a set of trigger channels for PPG1. When it is set to 1, TPU0 to TPU3 are selected as a set of trigger channels for PPG1.

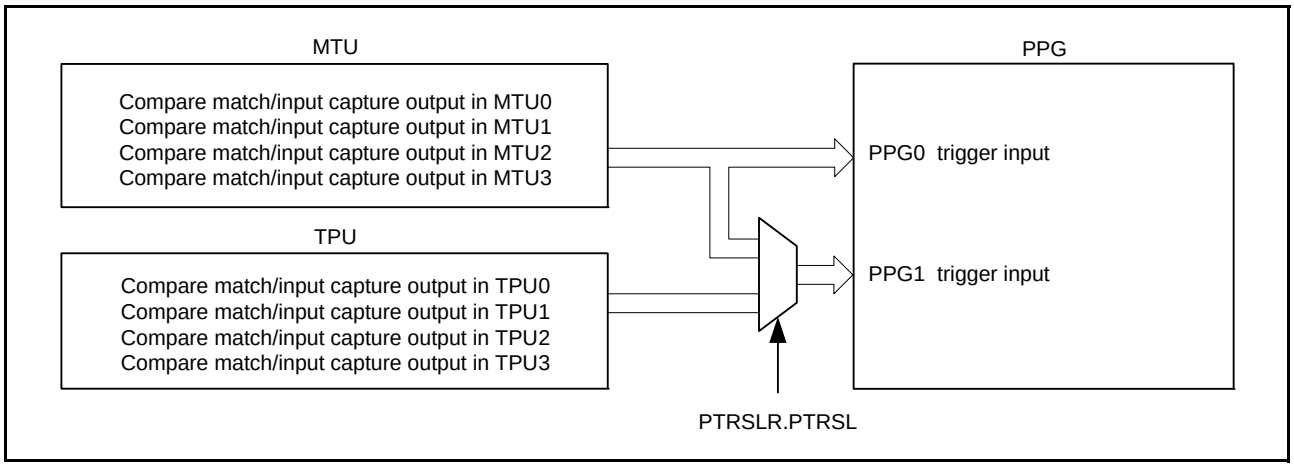


Figure 25.3 Block Diagram of PPG Trigger Selection

25.2.2 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.

- 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.

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.

- 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.

25.2.3 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.

25.2.4 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

25.2.5 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 25.2.6, PPG Output Mode Register (PMR).

25.2.6 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 MTU or TPU channel that functions as an output trigger.

25.3 Operation

Figure 25.4 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.

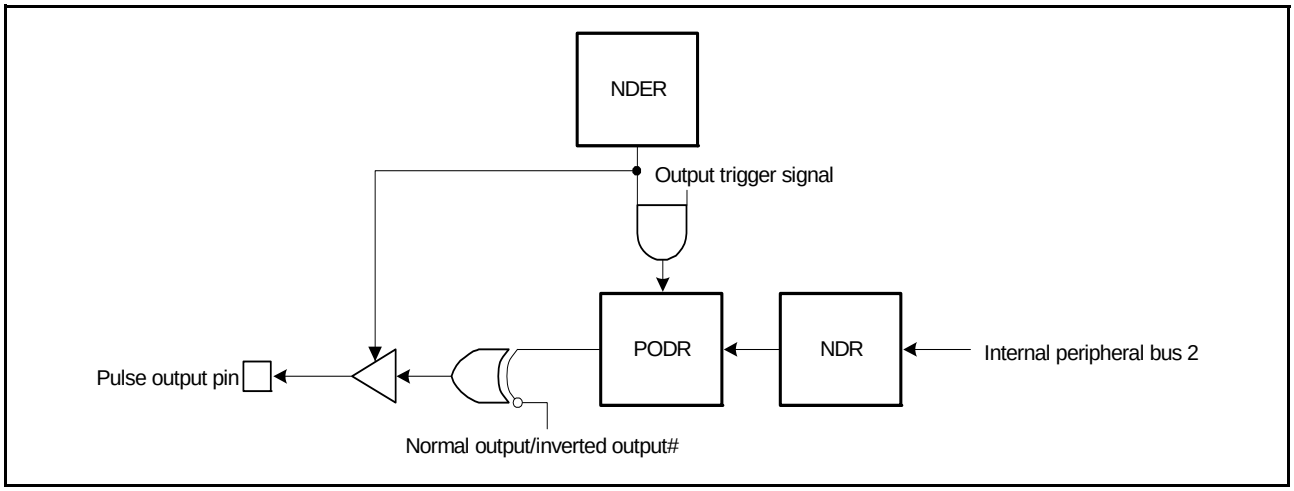


Figure 25.4 Schematic Diagram of PPG

25.3.1 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 25.5 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.

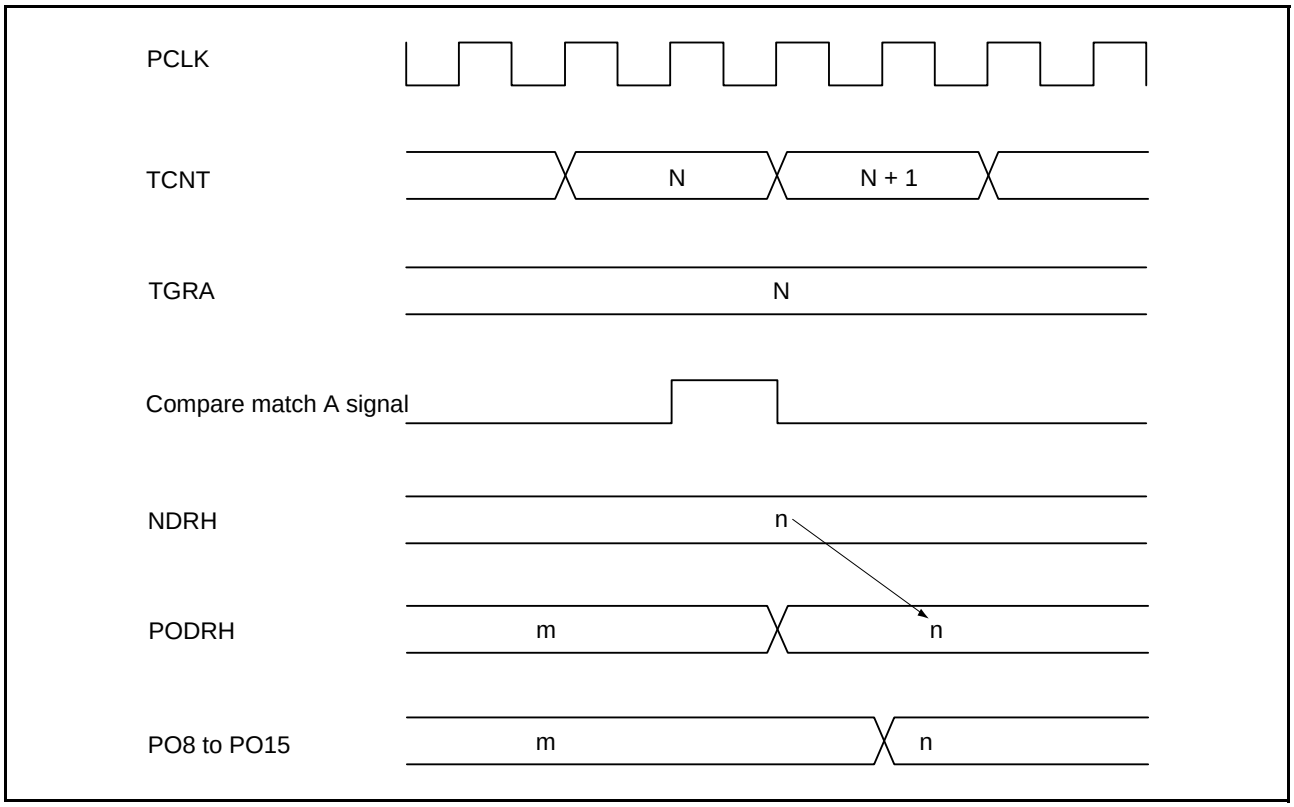


Figure 25.5 Timing of Transfer and Output of the Values in NDR (Example)

25.3.2 Sample Setup Procedure for Normal Pulse Output

Figure 25.6 and Figure 25.7 show sample procedures for setting normal pulse output.

(1) PPG0 Setting

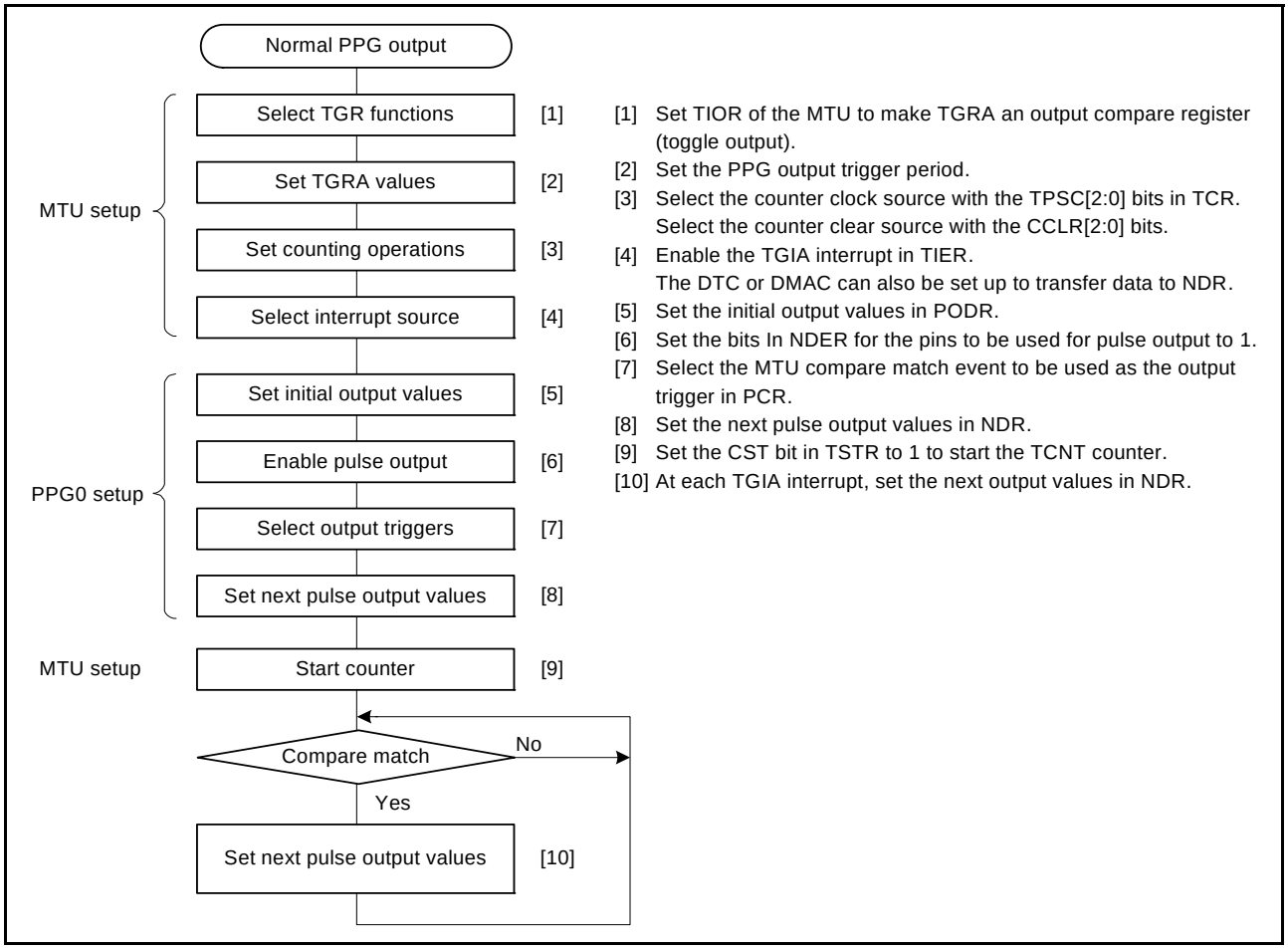


Figure 25.6 Sample Setup Procedure for Normal Pulse Output (PPG0 Setting)

(2) PPG1 Setting

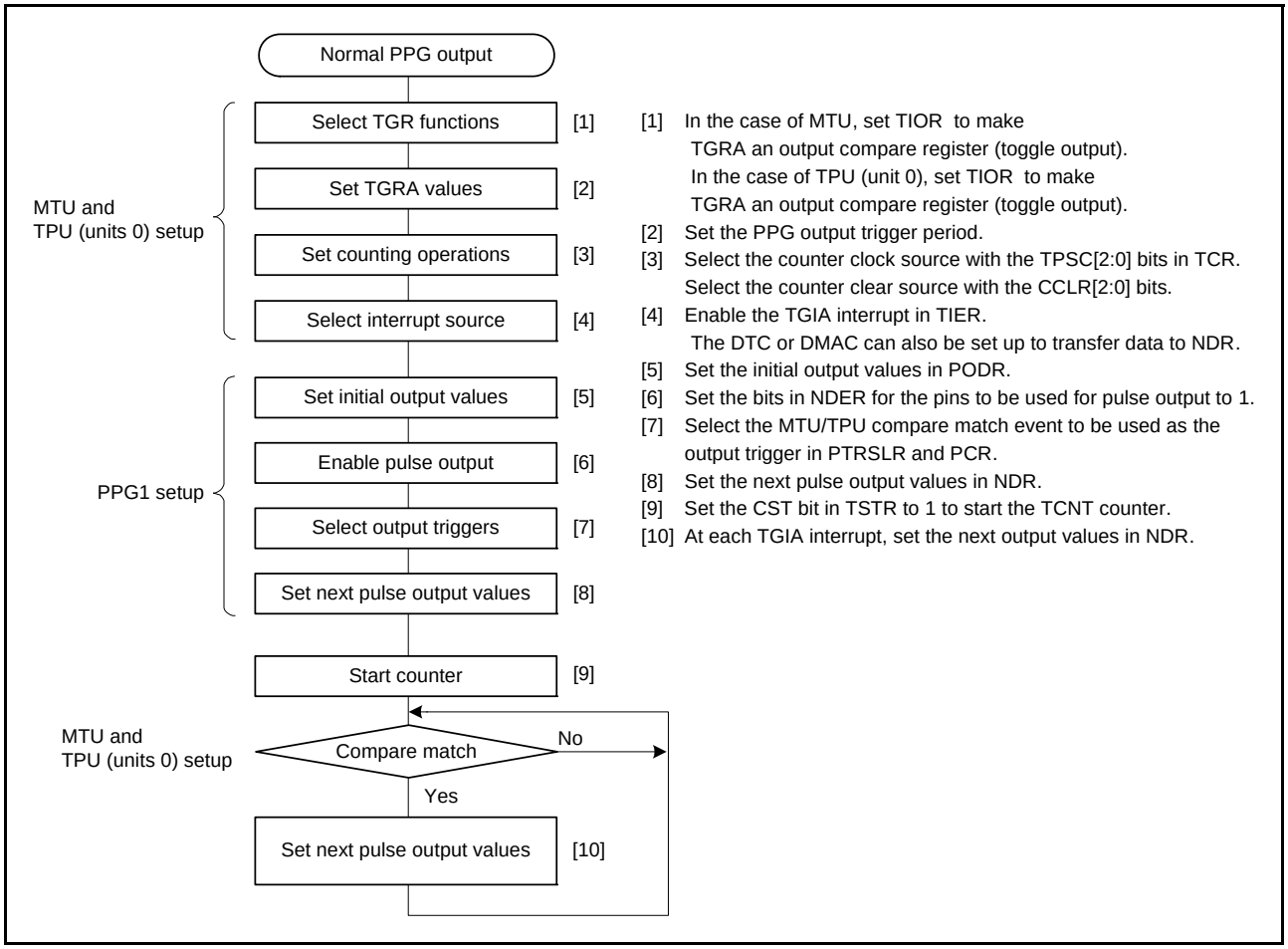


Figure 25.7 Sample Setup Procedure for Normal Pulse Output (PPG1 Setting)

25.3.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 25.8 shows an example in which pulse output from the PPG0 is used for cyclic five-phase pulse output.

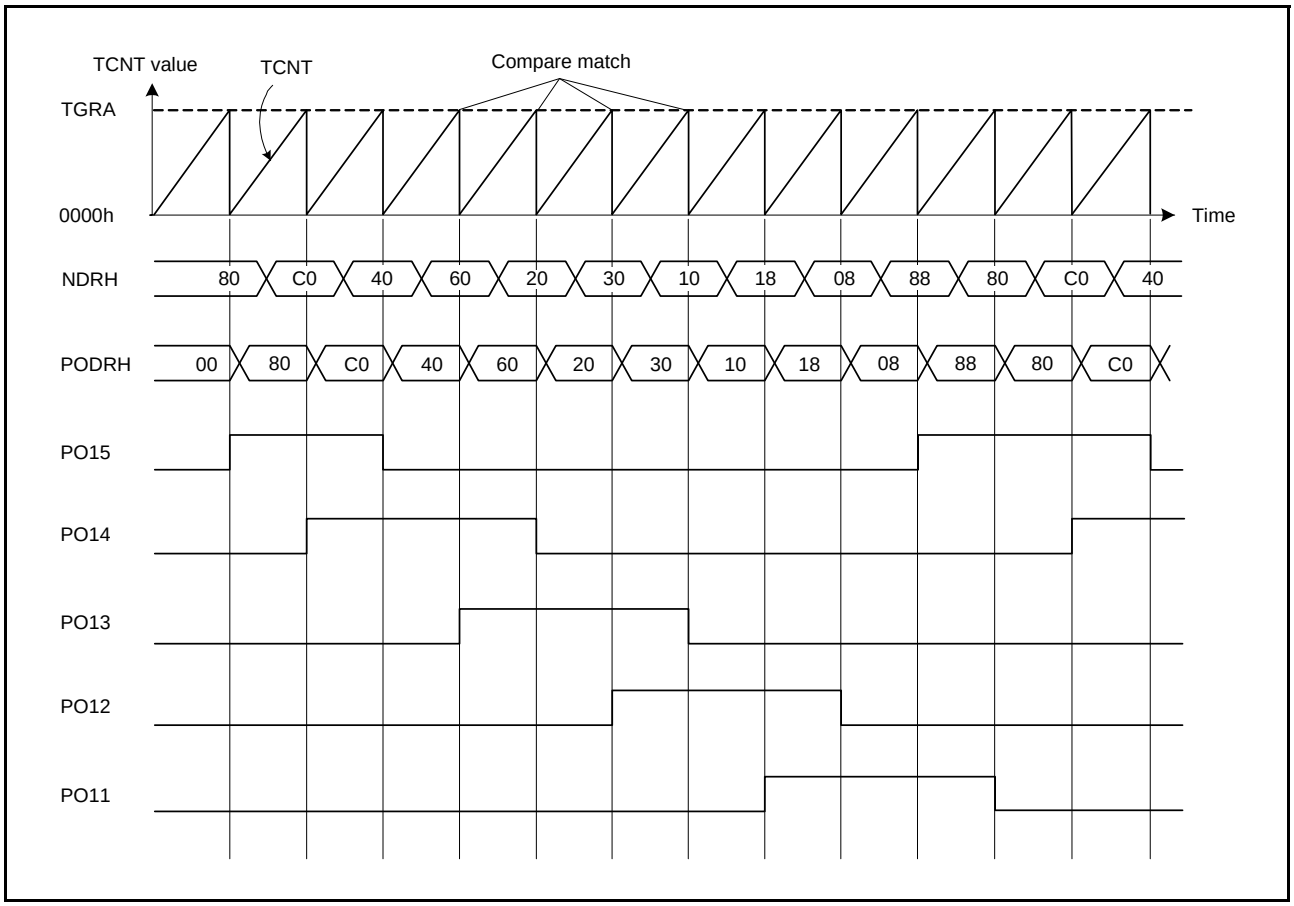


Figure 25.8 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

1. Set an output compare register of the MTUn.TGRA (n = 0 to 3) of MTU 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 MTU 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.

25.3.4 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 25.9 shows the non-overlapping pulse output operation.

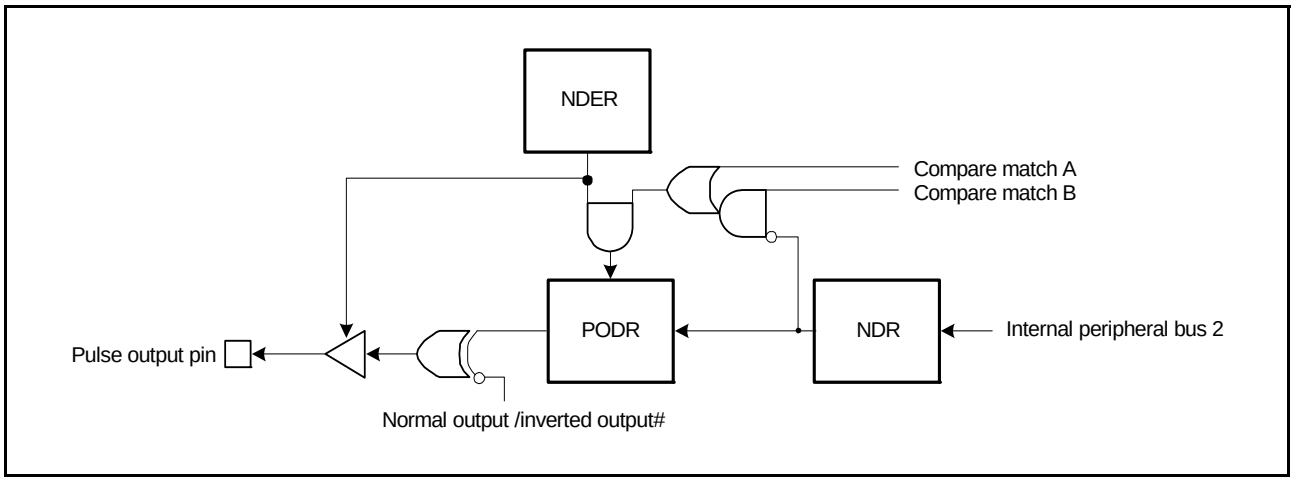


Figure 25.9 Non-Overlapping Pulse Output

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 25.10 shows the timing of the above procedure.

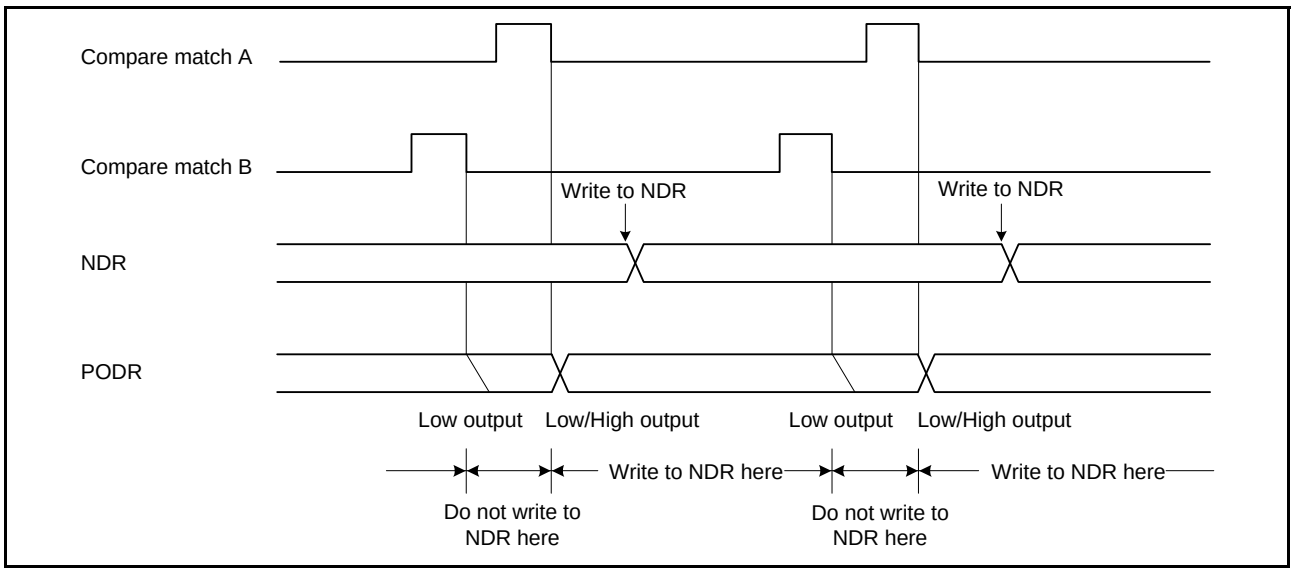


Figure 25.10 Non-Overlapping Operation and Write Timing to PPGn.NDRH and PPGn.NDRL

25.3.5 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 25.11 and Figure 25.12 show sample procedures for setting up non-overlapping pulse outputs.

(1) PPG0 Setting

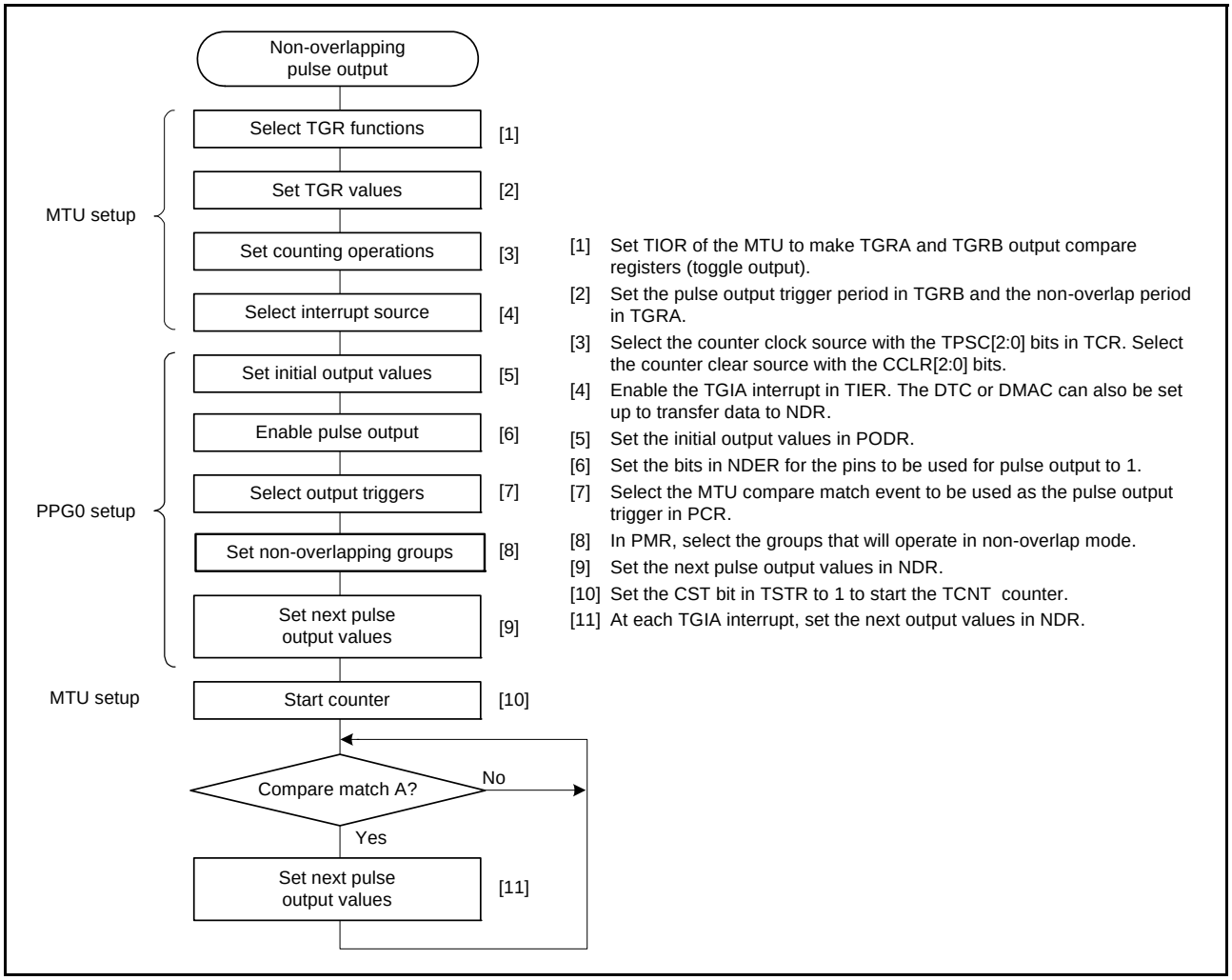


Figure 25.11 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG0 Setting)

(2) PPG1 Setting

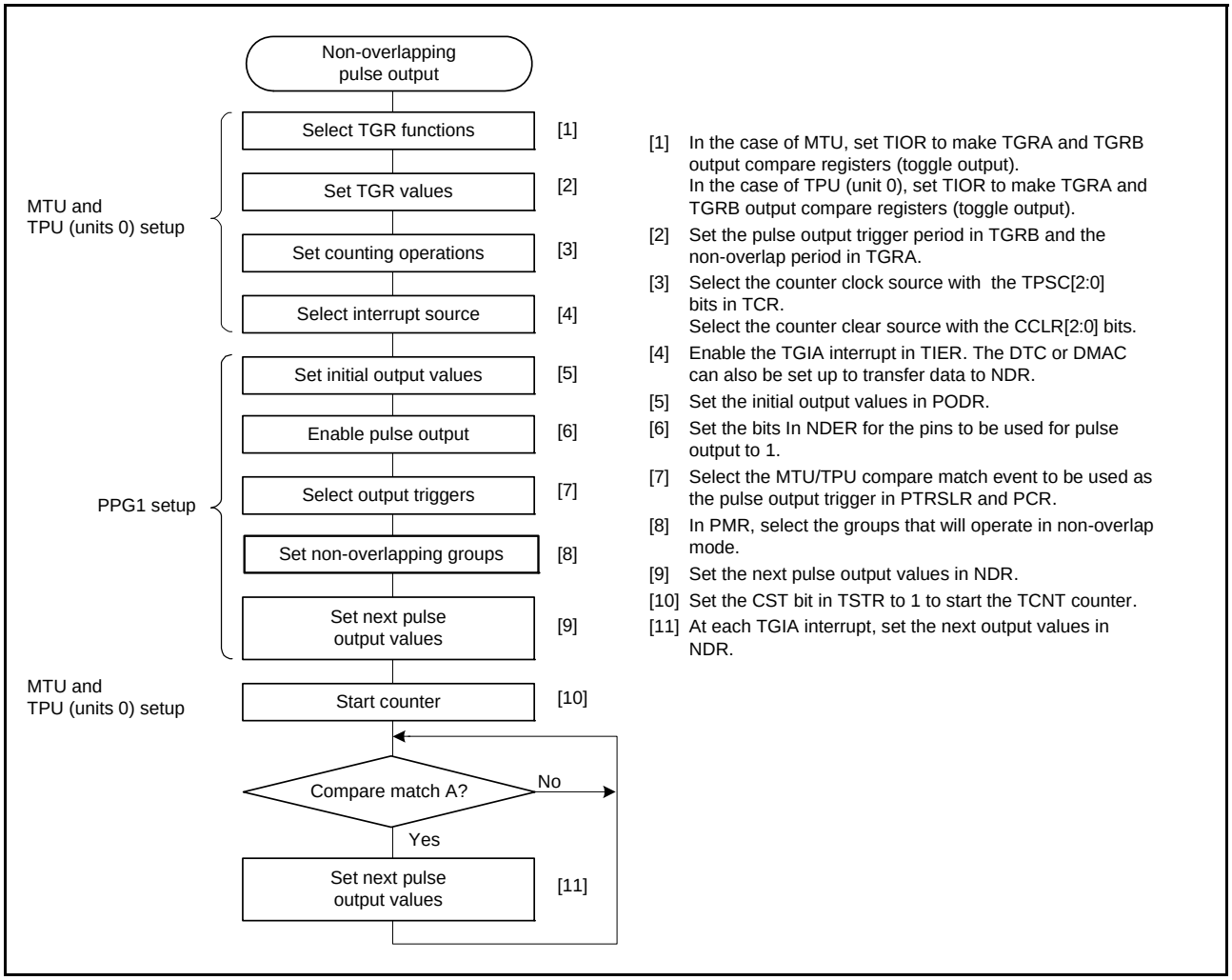


Figure 25.12 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG1 Setting)

25.3.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 25.13 shows an example in which pulse output from the PPG0 is used for four-phase complementary non-overlapping pulse output.

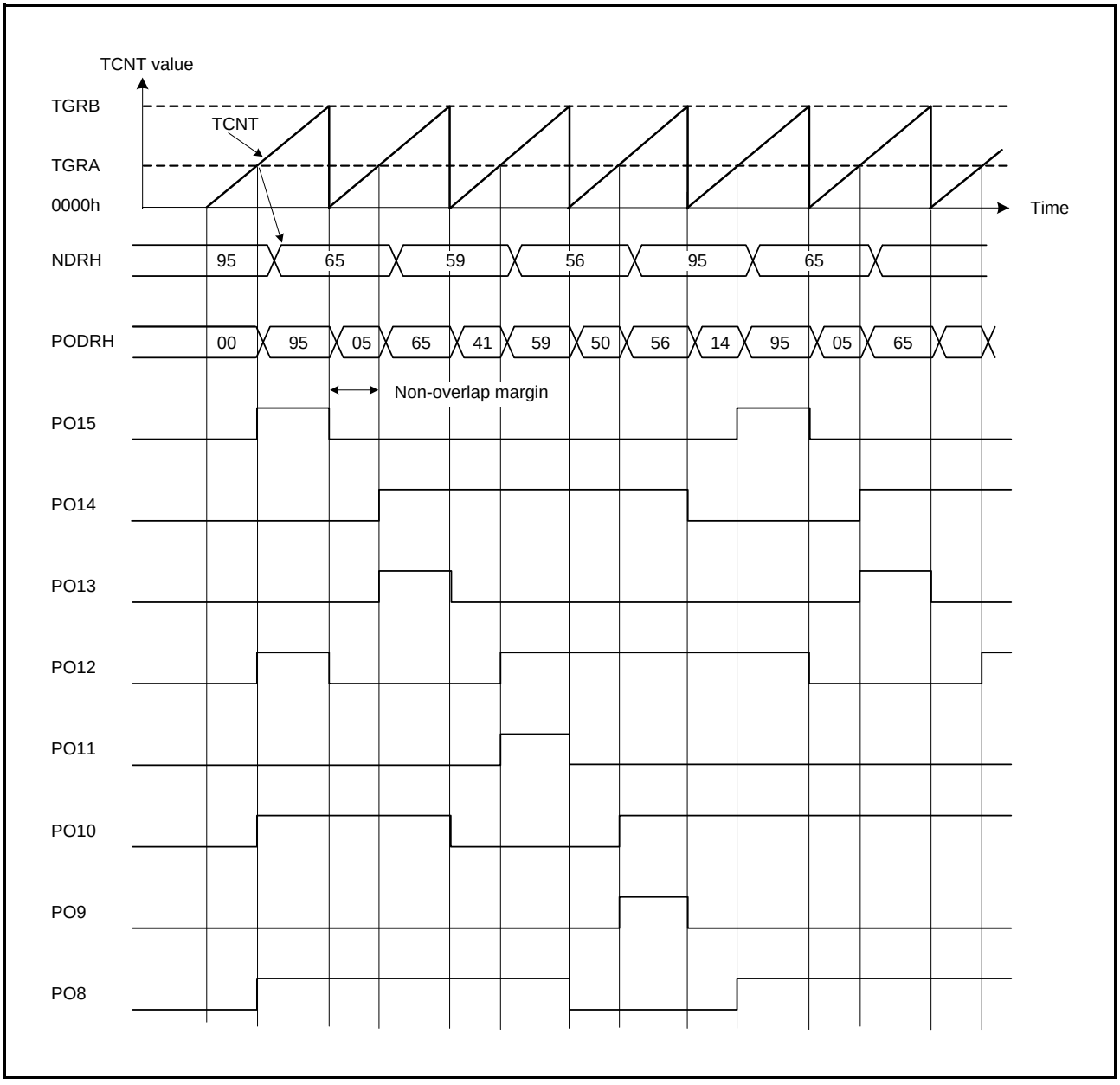


Figure 25.13 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

1. Set output compare registers of the MTUn.TGRA and MTUn.TGRB (n = 0 to 3) of MTU 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 MTU 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.

25.3.7 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 25.14 shows the outputs when the G3INV and G2INV bits are cleared to 0 in addition to the settings in Figure 25.13.

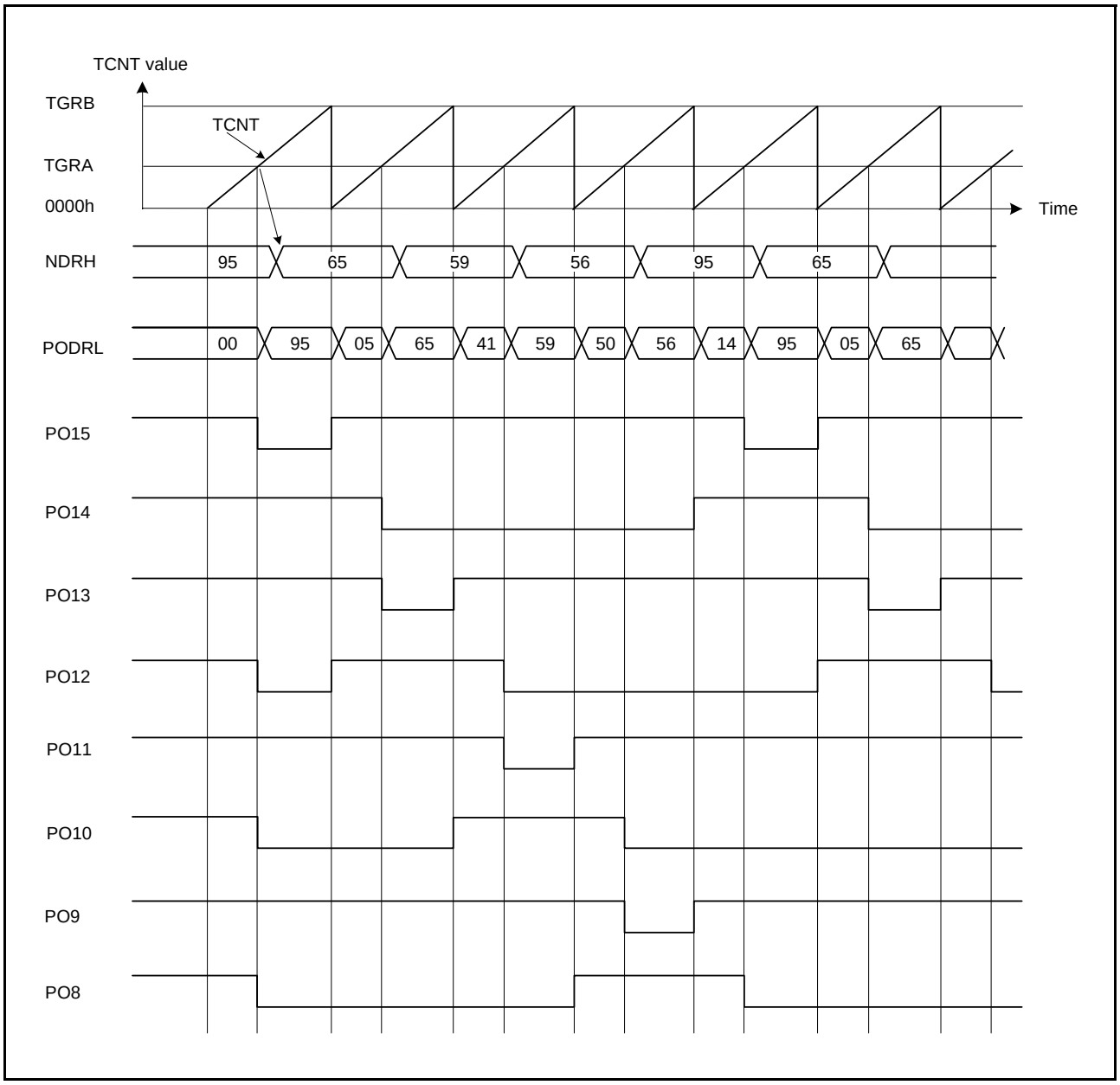


Figure 25.14 Inverted Pulse Output (Example)

25.3.8 Pulse Output Triggered by Input Capture

Pulse output from the PPG0 can be triggered by the MTU input capture as well as by compare match. When MTUn.TGRA (n = 0 to 3) functions as an input capture register in the MTU channel selected by PPG0.PCR, pulse output is triggered by the input capture signal.

Figure 25.15 shows the timing of pulse output triggered by input capture.

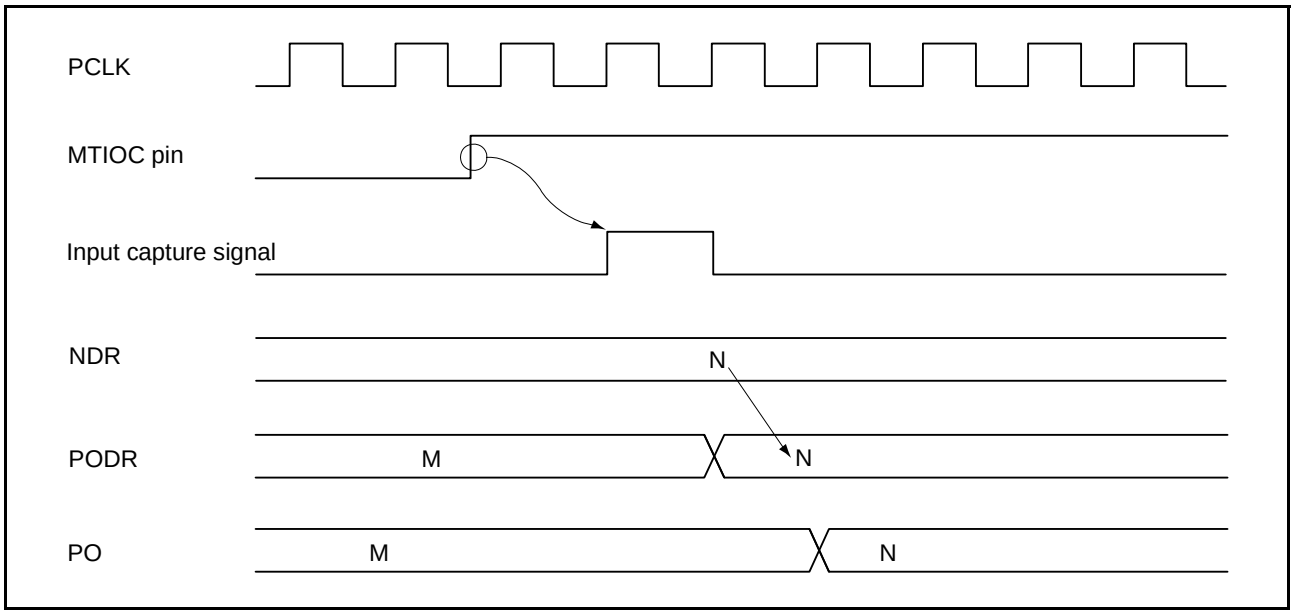


Figure 25.15 Timing of Pulse Output Triggered by Input Capture (Example)

25.4 Usage Note

25.4.1 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.

26. 8-Bit Timer (TMR)

The RX63N/RX631 Group 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 multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.
Unit 0 and unit 1 can generate a baud rate clock signal for the SCI and have the same functions.

26.1 Overview

Table 26.1 lists the specifications of the TMR.
Figure 26.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 26.2 shows that of the 8-bit timer module (unit 1).

Table 26.1 Specifications of TMR

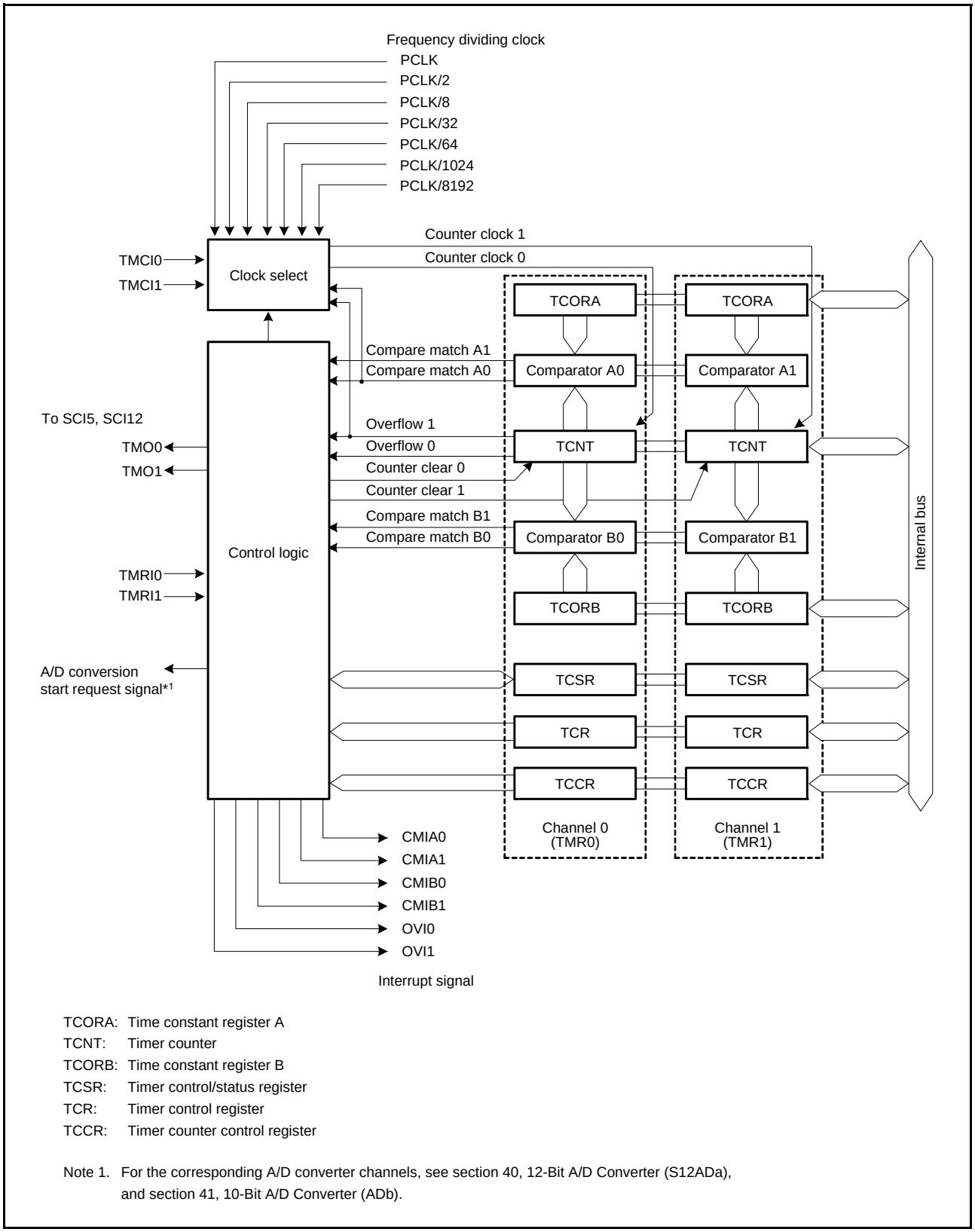
Item	Description
Count clock	• Frequency dividing clock: PCLK, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock
Number of channels	(8 bits × 2 channels) × 2 units
Compare match	• 8-bit mode (compare match A, compare match B) • 16-bit mode (compare match A, compare match B)
Counter clear	Selected by compare match A or B, or an external 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
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
Generation of trigger to start A/D converter conversion	Compare match A of TMR0 and TMR2*1
Capable of generating baud rate clock for SCI	Generates baud rate clock for SCI.*2
Low power consumption facilities	Each unit can be placed in a module stop state.

Note 1. For details, see section 40, 12-Bit A/D Converter (S12ADa), and section 41, 10-Bit A/D Converter (ADb).
Note 2. For details, see section 34, Serial Communications Interface (SCIc, SCId).

Table 26.2 Pin Configuration of TMR

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 PCLK2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI0	PCLK/1 PCLK2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI2	PCLK/1 PCLK2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3	PCLK/1 PCLK2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3
Counter clear		TMR0.TCORA TMR0.TCORAB TMRI0	TMR0.TCORA TMR0.TCORAB TMRI1	TMR0.TCORA + TMR1.TCORA TMR0.TCORB+ TMR1.TCORB TMRI0	TMR2.TCORA TMR2.TCORAB TMRI2	TMR3.TCORA TMR3.TCORAB TMRI1	TMR2.TCORA + TMR3.TCORA TMR2.TCORB+ TMR3.TCORB TMRI0
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 converter conversion start trigger*1		○	—	○	○	—	○
SCI baud rate clock generation*2		○		—	○		—
Module stop setting*3		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

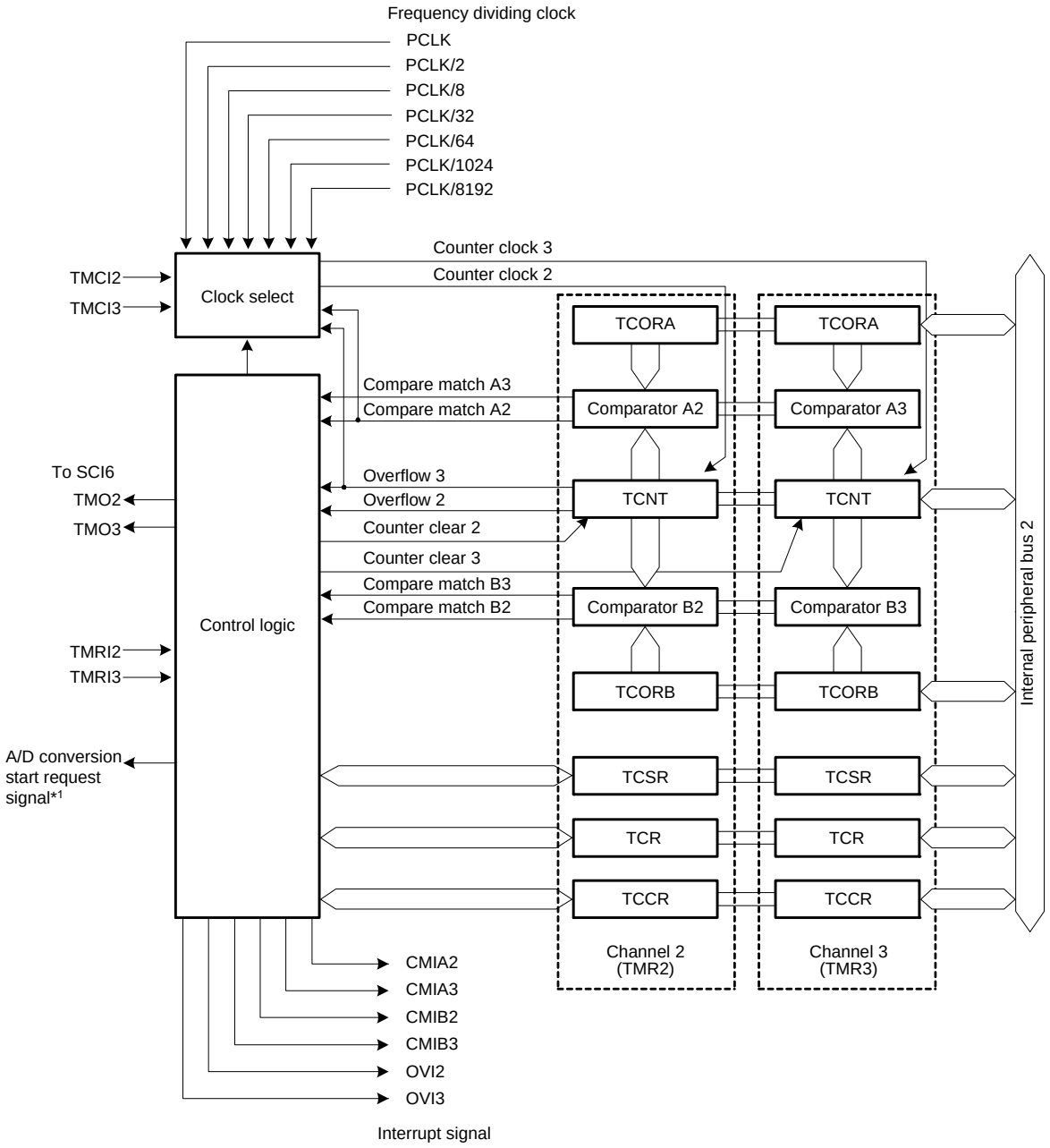
○: Possible
—: Impossible
Note 1. For details, see section 40, 12-Bit A/D Converter (S12ADa), and section 41, 10-Bit A/D Converter (ADb).
Note 2. For details, see section 34, Serial Communications Interface (SCIc, SCId).
Note 3. For details, see section 11, Low Power Consumption.



TCORA: Time constant register A
TCNT: Timer counter
TCORB: Time constant register B
TCSR: Timer control/status register
TCR: Timer control register
TCCR: Timer counter control register

Note 1. For the corresponding A/D converter channels, see section 40, 12-Bit A/D Converter (S12ADa), and section 41, 10-Bit A/D Converter (ADb).

Figure 26.1 Block Diagram of TMR (Unit 0)



TCORA: Time constant register A
TCNT: Timer counter
TCORB: Time constant register B
TCSR: Timer control/status register
TCR: Timer control register
TCCR: Timer counter control register

Note 1. For the corresponding A/D converter channels, see section 40, 12-Bit A/D Converter (S12ADa), and section 41, 10-Bit A/D Converter (ADb).

Figure 26.2 Block Diagram of TMR (Unit 1)

Table 26.3 lists the input/output pins of the TMR.

Table 26.3 Pin Configuration of TMR

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMCI0	Input	Inputs external clock for counter
		TMRI0	Input	Inputs external reset to counter
	TMR1	TMO1	Output	Outputs compare match
		TMCI1	Input	Inputs external clock for counter
		TMRI1	Input	Inputs external reset to counter
1	TMR2	TMO2	Output	Outputs compare match
		TMCI2	Input	Inputs external clock for counter
		TMRI2	Input	Inputs external reset to counter
	TMR3	TMO3	Output	Outputs compare match
		TMCI3	Input	Inputs external clock for counter
		TMRI3	Input	Inputs external reset to counter

26.2 Register Descriptions

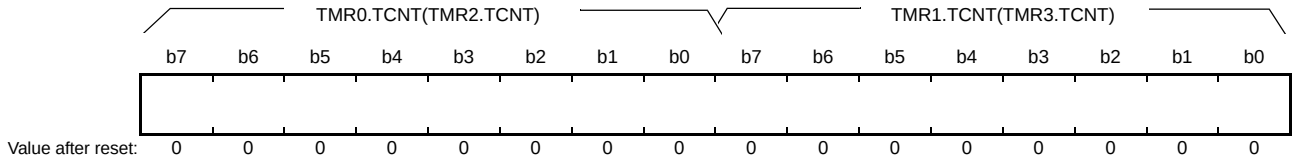
Table 26.4 lists register allocation for 16-bit access.

Table 26.4 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

26.2.1 Timer Counter (TCNT)

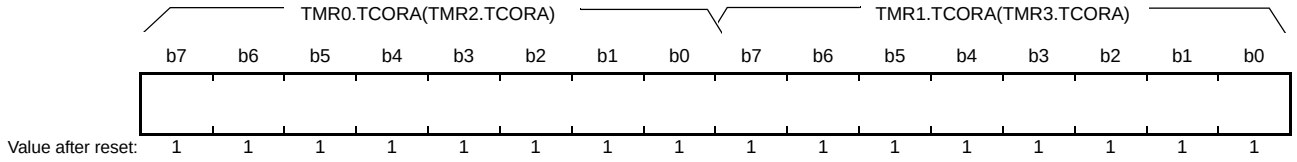
Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h
TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h



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 clock.
TCNT can be cleared by an external reset input signal, compare match A signal, or compare match B signal. Which signal to be used for clearing is selected by the TCR.CCLR[1:0] bits.
When TCNT overflows from FFh to 00h, the interrupt flag is set to 1.
For details on the corresponding interrupt vector number, see section 15, Interrupt Controller (ICUb), and Table 26.6, TMR Interrupt Sources.

26.2.2 Time Constant Register A (TCORA)

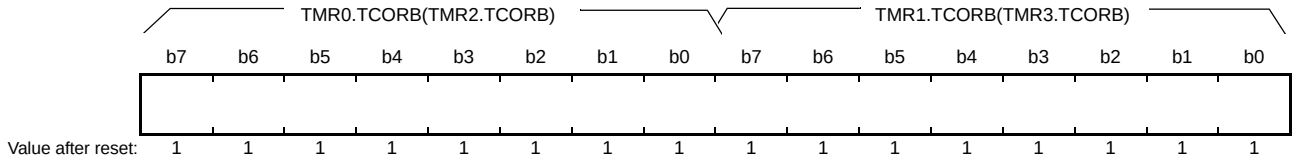
Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h
TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h



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 signal is set to high. 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 signal and the settings of the TCSR.OSA[1:0] bits.

26.2.3 Time Constant Register B (TCORB)

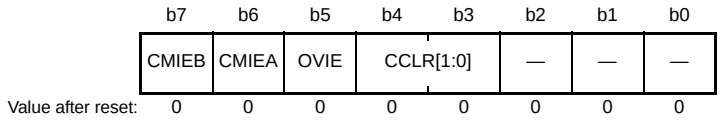
Address(es): TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h
TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h



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 signal is set to high. 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 signal and the settings of the TCSR.OSB[1:0] bits.

26.2.4 Timer Control Register (TCR)

Address(es): TMR0.TCR 0008 8200h, TMR1.TCR 0008 8201h
TMR2.TCR 0008 8210h, TMR3.TCR 0008 8211h



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 reset input (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 reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

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.

26.2.5 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 26.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 26.5.	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 reset 1: Cleared when the external reset is high	R/W

Note 1. To use an external reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

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 26.5.

TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external reset input) and selects the condition for detecting external reset (level or edge).

Table 26.5 Clock Input to TCNT and Count Condition

Channel	TCCR Register					Description
	CSS[1:0]		CKS[2:0]			
	b4	b3	b2	b1	b0	
TMR0 (TMR2)	0	0	—	0	0	Clock input prohibited
					1	Uses external clock. Counts at rising edge*1.
				1	0	Uses external clock. Counts at falling edge*1.
					1	Uses external clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses frequency dividing clock. Counts at PCLK.
					1	Uses frequency dividing clock. Counts at PCLK/2.
				1	0	Uses frequency dividing clock. Counts at PCLK/8.
					1	Uses frequency dividing clock. Counts at PCLK/32.
			1	0	0	Uses frequency dividing clock. Counts at PCLK/64.
					1	Uses frequency dividing clock. Counts at PCLK/1024.
				1	0	Uses frequency dividing 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 clock. Counts at rising edge*1.
				1	0	Uses external clock. Counts at falling edge*1.
					1	Uses external clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses frequency dividing clock. Counts at PCLK.
					1	Uses frequency dividing clock. Counts at PCLK/2.
				1	0	Uses frequency dividing clock. Counts at PCLK/8.
					1	Uses frequency dividing clock. Counts at PCLK/32.
			1	0	0	Uses frequency dividing clock. Counts at PCLK/64.
					1	Uses frequency dividing clock. Counts at PCLK/1024.
				1	0	Uses frequency dividing 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 reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

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 incrementing clock is generated. Do not use this setting.

26.2.6 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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change when compare match A occurs 0 1: Low is output when compare match A occurs 1 0: High is output when compare match A occurs 1 1: Output is inverted when compare match A occurs (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change when compare match B occurs 0 1: Low is output when compare match B occurs 1 0: High is output when compare match B occurs 1 1: Output is inverted when compare match B occurs (toggle output)	R/W
b4	ADTE	A/D Trigger Enable*2	0: A/D converter start requests by compare match A are disabled 1: A/D converter start requests by compare match A are 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. If the output enable signal for the corresponding TMOn pin is negated while all of the OSA[1:0] and OSB[1:0] bits are 0, a request for high-impedance output is issued for the given I/O port pin. If any of the OSA[1:0] or OSB[1:0] bits is 1, the timer output remains at the low level until the first compare match after a reset.

Note 2. For the corresponding A/D converter channels, see section 40, 12-Bit A/D Converter (S12ADa), and section 41, 10-Bit A/D Converter (ADb).

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change when compare match A occurs 0 1: Low is output when compare match A occurs 1 0: High is output when compare match A occurs 1 1: Output is inverted when compare match A occurs (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change when compare match B occurs 0 1: Low is output when compare match B occurs 1 0: High is output when compare match B occurs 1 1: Output is inverted when compare match B occurs (toggle output)	R/W
b4	—	Reserved	This bit is read as 1. The write value should always 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. If the output enable signal for the corresponding TMOn pin is negated while all of the OSA[1:0] and OSB[1:0] bits are 0, a request for high-impedance output is issued for the given I/O port pin. If any of the OSA[1:0] or OSB[1:0] bits is 1, the timer output remains at the low level until the first compare match after a reset.

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.

26.3 Operation

26.3.1 Pulse Output

Figure 26.3 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 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.

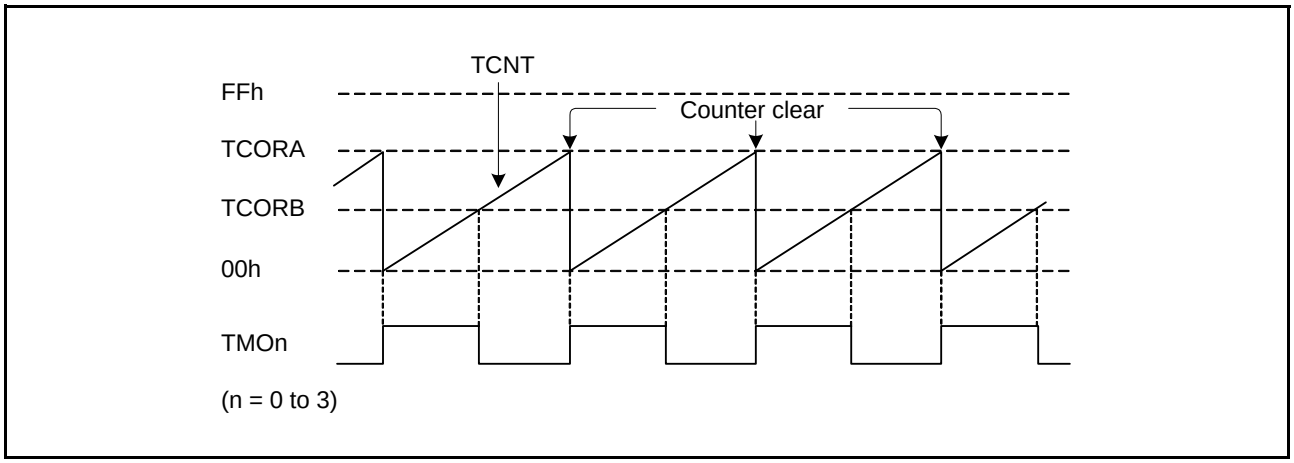


Figure 26.3 Example of Pulse Output

26.3.2 Reset Input

Figure 26.4 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 reset input) and set the TMRIS bit in TCCR to 1 (cleared when the external reset 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.

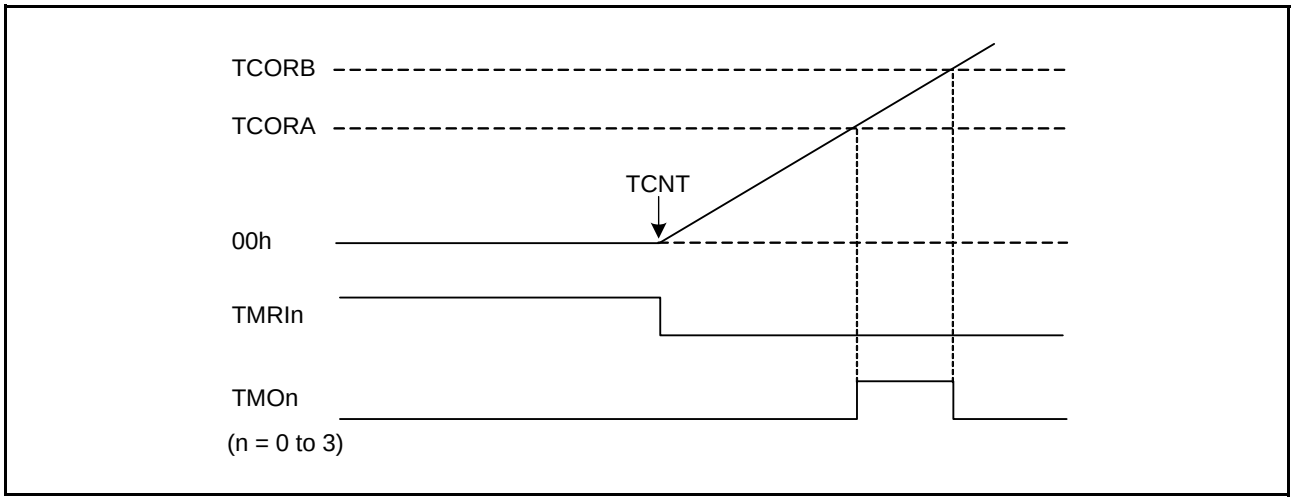


Figure 26.4 Example of Reset Input

26.4 Operation Timing

26.4.1 TCNT Count Timing

Figure 26.5 shows the count timing of TCNT for frequency dividing clock input. Figure 26.6 shows the count timing of TCNT for external clock input.

Note that the external clock pulse width must be at least 1.5 or more PCLK cycles for increment at a single edge, and at least 2.5 or more PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

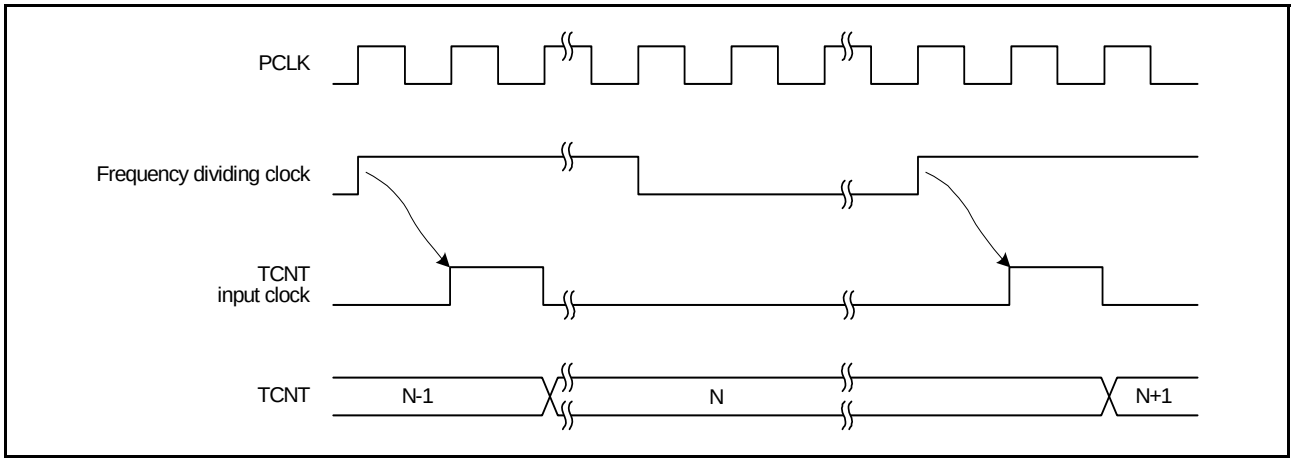


Figure 26.5 Count Timing for Frequency Dividing Clock Input

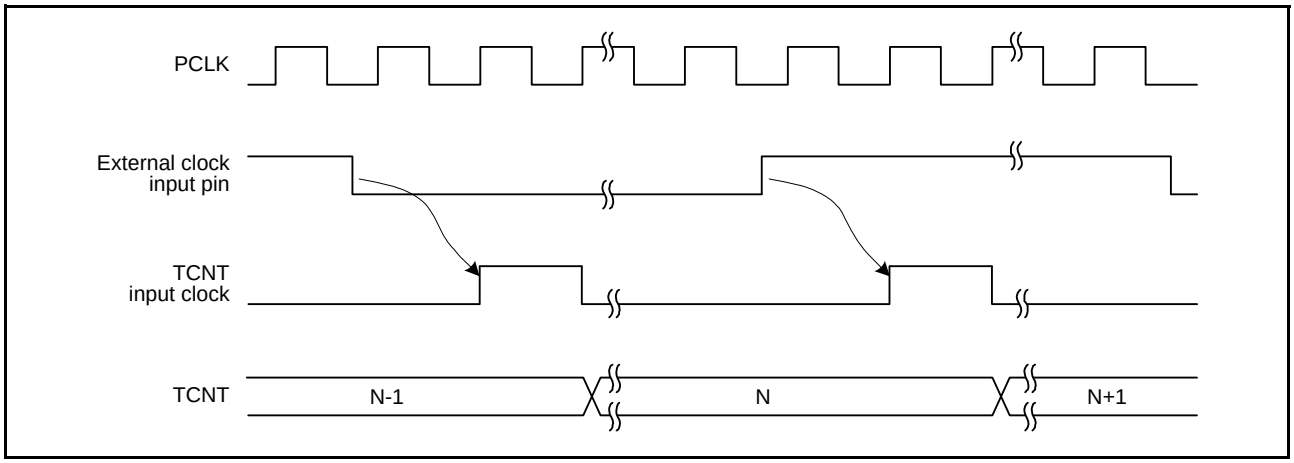


Figure 26.6 Count Timing for External Clock Input (at Both Edges)

26.4.2 Timing of Interrupt Flag Setting to 1 at Compare Match

The interrupt flag is set to 1 by a compare match signal generated when values of TCORA and TCORB and that of TCNT match.

The compare match signal is generated at the last PCLK cycle in which the match is true, just before the timer counter is updated. Therefore, when values of TCORA and TCORB and that of TCNT match, the compare match signal is not generated until the next TCNT clock input.

Figure 26.7 shows this timing.

For details on the corresponding interrupt vector number, see section 15, Interrupt Controller (ICUb), and Table 26.6, TMR Interrupt Sources.

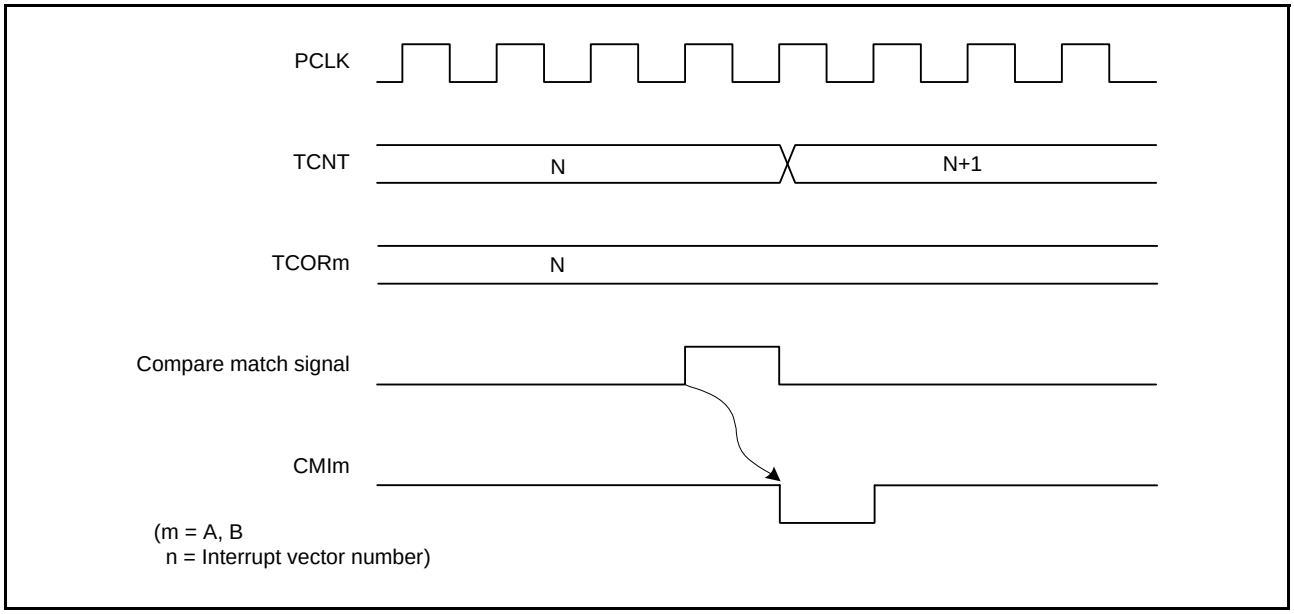


Figure 26.7 Timing of Interrupt Flag Setting to 1 at Compare Match

26.4.3 Timing of Timer Output at Compare Match

When a compare match signal is generated, the timer output changes as specified by the TCSR.OSA[1:0] and OSB[1:0] bits.

Figure 26.8 shows the timing when the timer output is toggled by the compare match A signal.

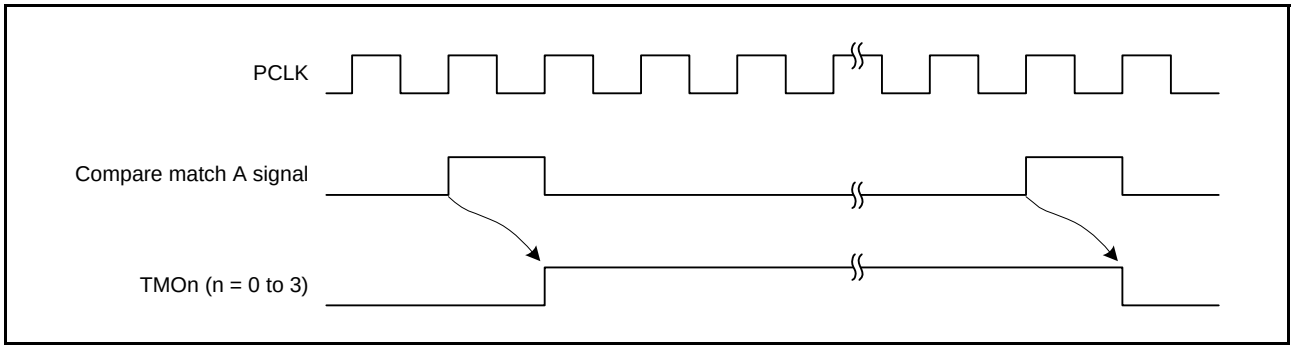


Figure 26.8 Timing of Timer Output at Compare Match A

26.4.4 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 26.9 shows the timing of this operation.

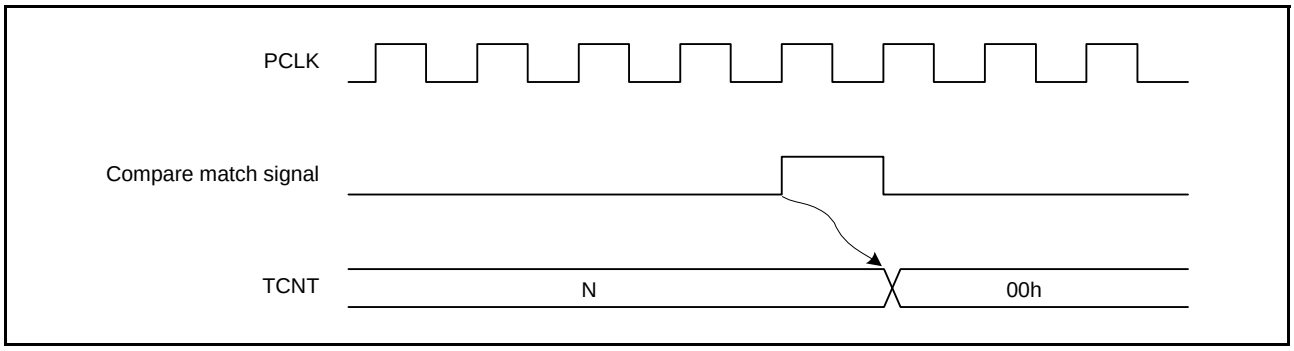


Figure 26.9 Timing of Counter Clear by Compare Match

26.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external reset input, depending on the settings of the TCR.CCLR[1:0] bits. At least two or more PCLK cycles are required from an external reset input to clearing of TCNT. Figure 26.10 and Figure 26.11 show the timing of this operation.

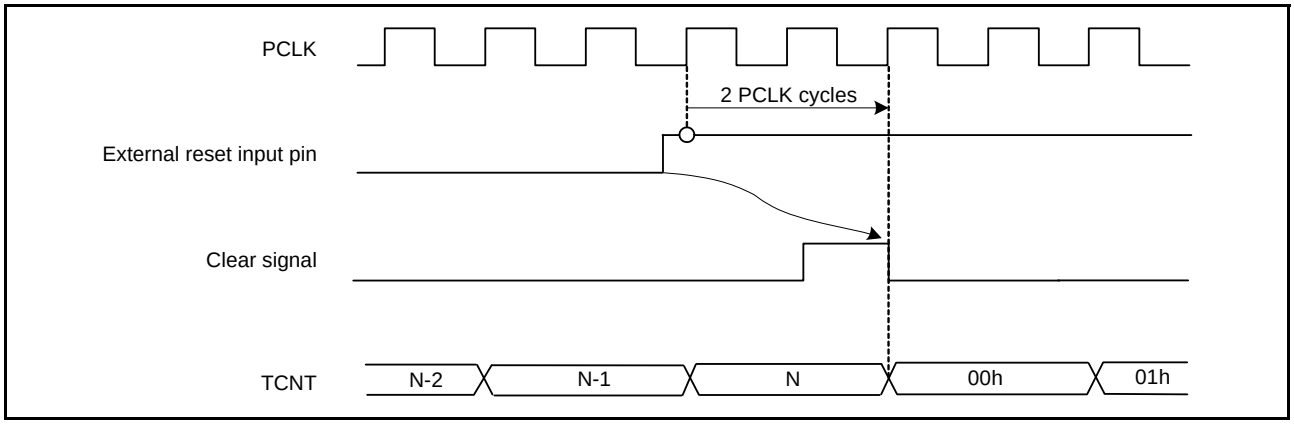


Figure 26.10 Timing of Clearance by External Reset (Rising Edge)

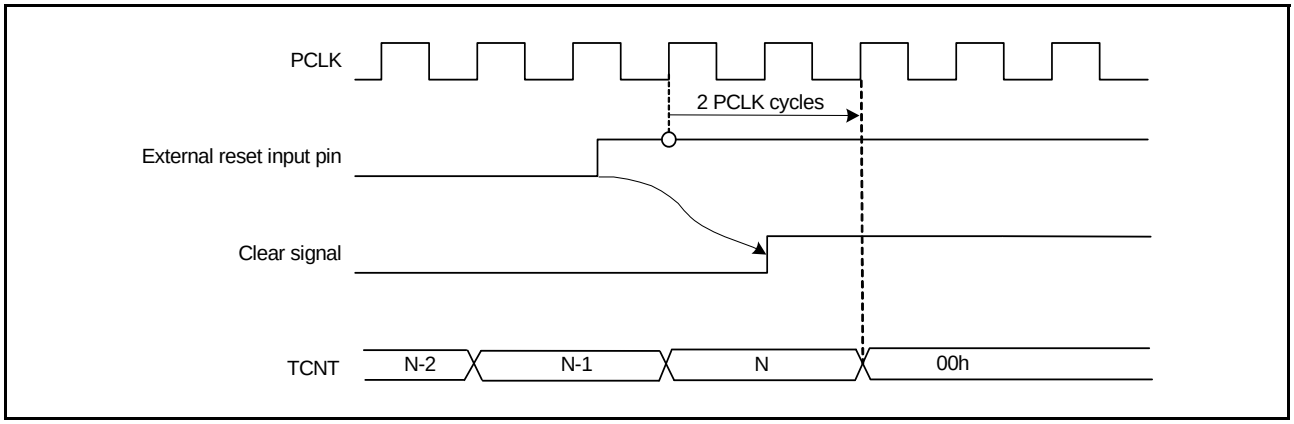


Figure 26.11 Timing of Clearance by External Reset (High Level)

26.4.6 Timing of Overflow Interrupt Flag Setting to 1

The interrupt flag is set to 1 by an overflow signal output when TCNT overflows (changes from FFh to 00h).
Figure 26.12 shows the timing of this operation.
For details on the corresponding interrupt vector number, see section 15, Interrupt Controller (ICUb), and Table 26.6, TMR Interrupt Sources.

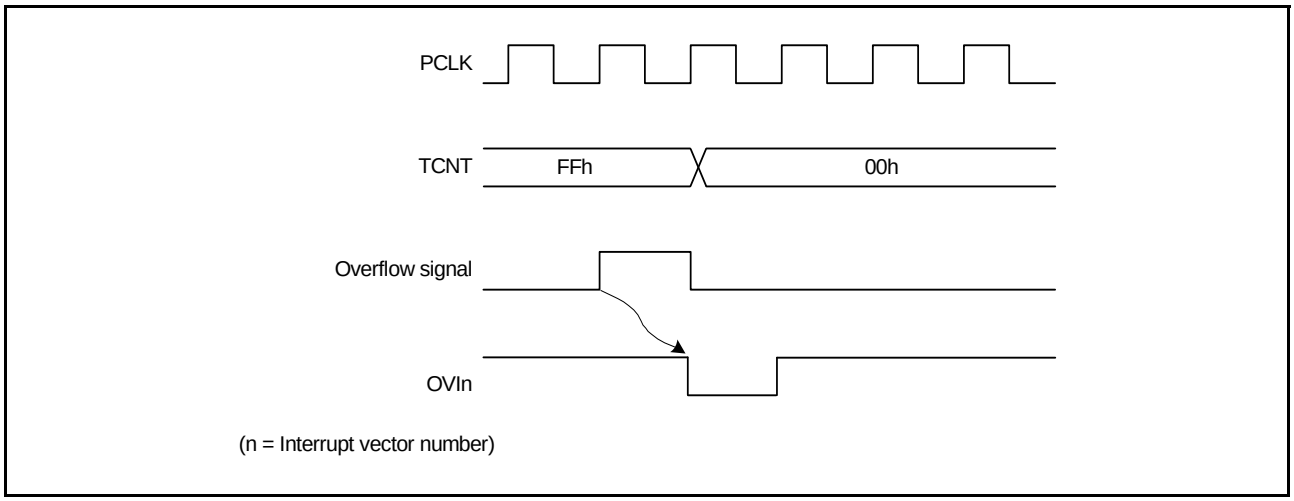


Figure 26.12 Timing of Overflow Interrupt Flag Setting to 1

26.5 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.

26.5.1 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.

26.5.2 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.

26.6 Interrupt Sources

26.6.1 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 26.6.

It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts. The DMAC cannot be activated by the interrupt sources for TMRn.

Table 26.6 TMR Interrupt Sources

Name	Interrupt Sources	DTC Activation	Priority
CMIA0	TMR0.TCORA compare match	Possible	High
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	Low

26.6.2 A/D Converter Activation

The A/D converter*1 can be activated by a compare match A for TMR0 or TMR2.

If the TMRn.TCSR.ADTE bit is set to 1 (A/D converter start requests by compare match A are enabled), a request to start A/D conversion is sent to the A/D converter by the occurrence of a compare match A. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

Note 1. For the corresponding A/D converter units, see section 40, 12-Bit A/D Converter (S12ADa), and section 41, 10-Bit A/D Converter (ADb).

Table 26.7 A/D Converter Activation

Module Symbol	Unit	Target	The Source of A/D Activation	A/D Conversion Start Request
S12AD	0	TMR0.TCORA, TMR0.TCNT	Compare match	TMTRG0AN_0
	1	TMR2.TCORA, TMR2.TCNT		TMTRG0AN_1
AD	0	TMR0.TCORA, TMR0.TCNT		TMTRG0AN_0

26.7 Usage Notes

26.7.1 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.

26.7.2 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).

$$f = PCLK / (N + 1)$$

26.7.3 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 26.13.

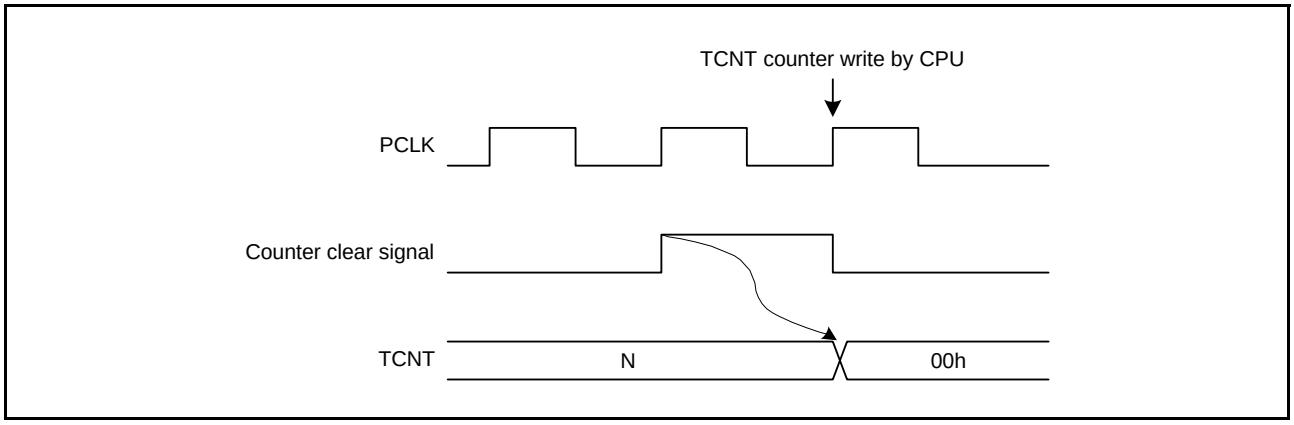


Figure 26.13 Conflict between TCNT Write and Counter Clear

26.7.4 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 26.14.

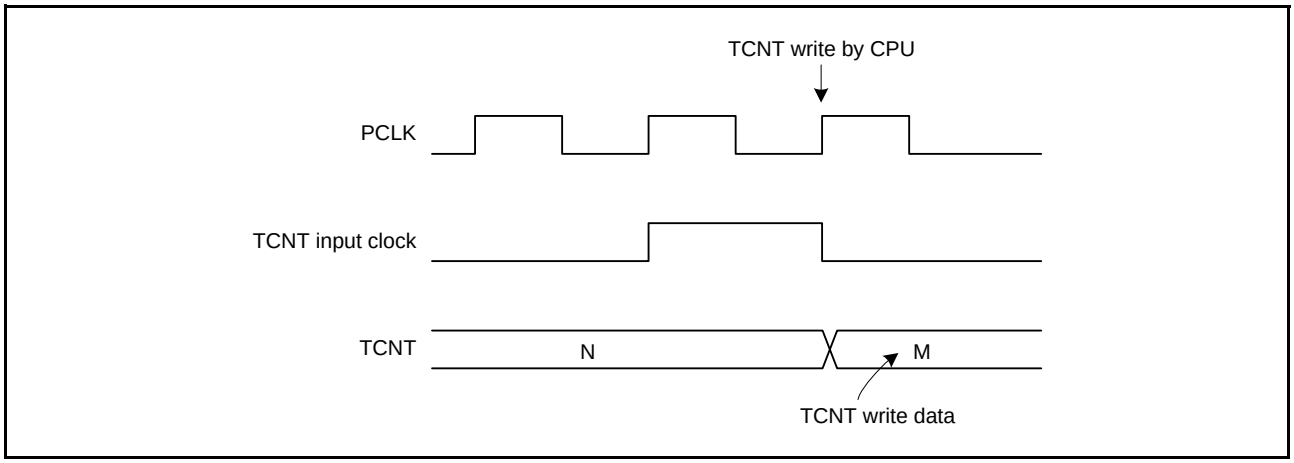


Figure 26.14 Conflict between TCNT Write and Increment

26.7.5 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, the write takes priority and the compare match signal is not high as shown in Figure 26.15.

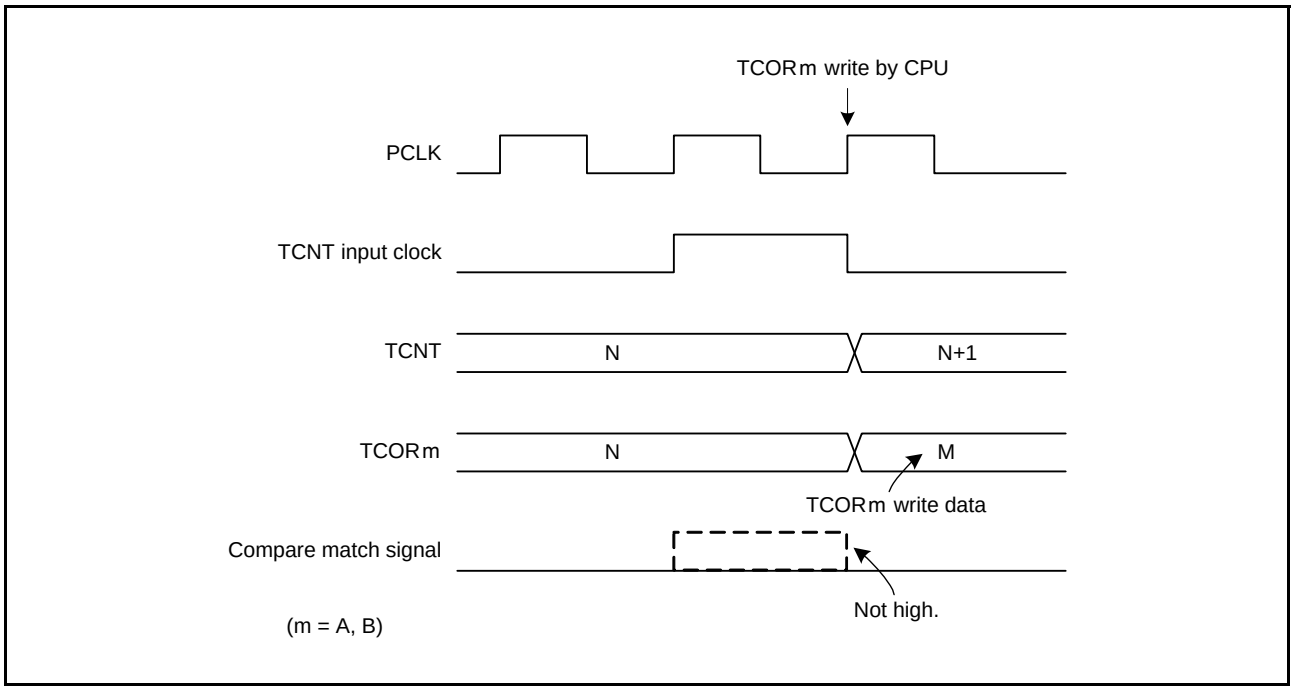


Figure 26.15 Conflict between TCORA or TCORB Write and Compare Match

26.7.6 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 statuses high for compare match A and compare match B, as listed in Table 26.8.

Table 26.8 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	↓
No change	Low

26.7.7 Switching of Frequency Dividing Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the frequency dividing clock is switched. Table 26.9 lists the relationship between the timing at which the frequency dividing clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

When TCNT clock is generated from an frequency dividing clock, the rising edge of the frequency dividing 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 26.9, the change is considered as an edge. Therefore, a TCNT clock pulse is generated and TCNT is incremented.

The erroneous increment of TCNT can also happen when switching between external and frequency dividing clocks.

Table 26.9 Switching of Frequency Dividing Clocks and TCNT Operation (1/2)

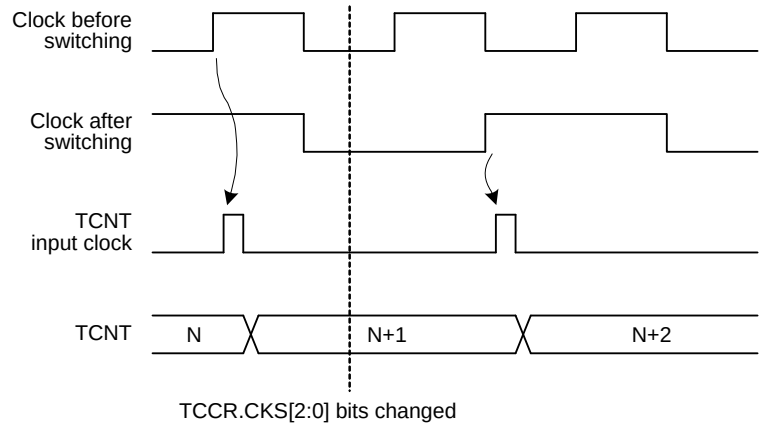
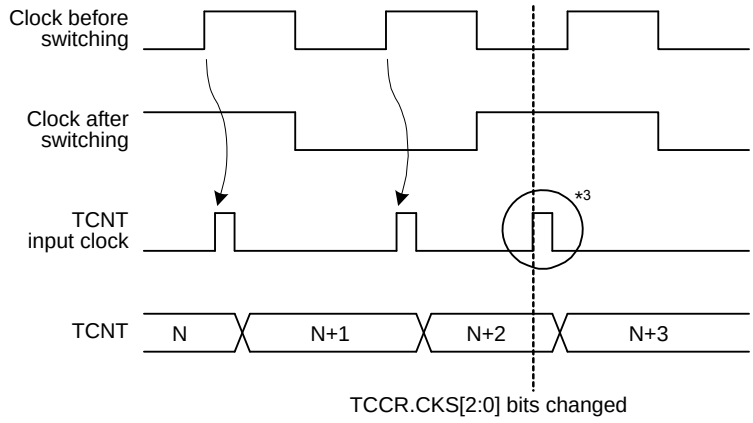
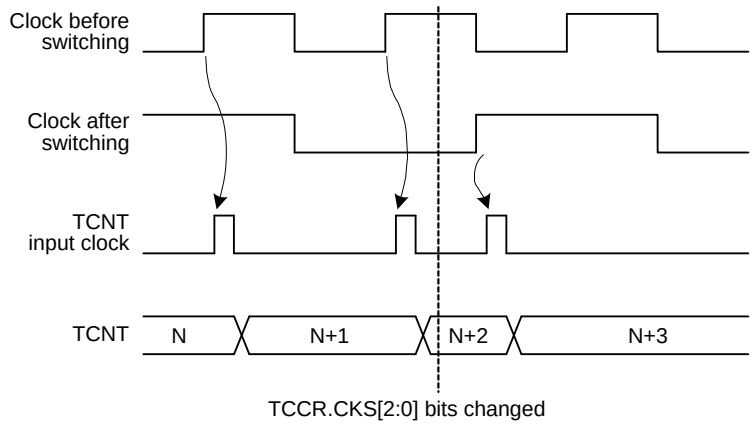
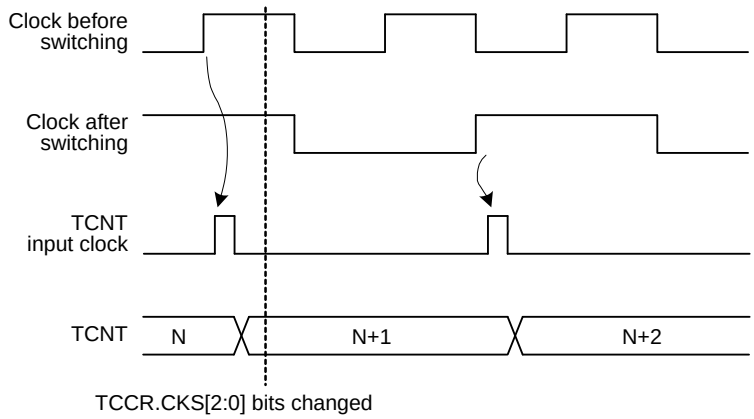
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Clock Operation
1	Switching from low to low*1	

Table 26.9 Switching of Frequency Dividing Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Clock Operation
2	Switching from low to high*2	
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 is incremented.
Note 4. Includes switching from high to stop.

26.7.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input 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.

26.7.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the frequency dividing 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 26.16 shows operation timing when the compare match interrupt signal is continuously output.

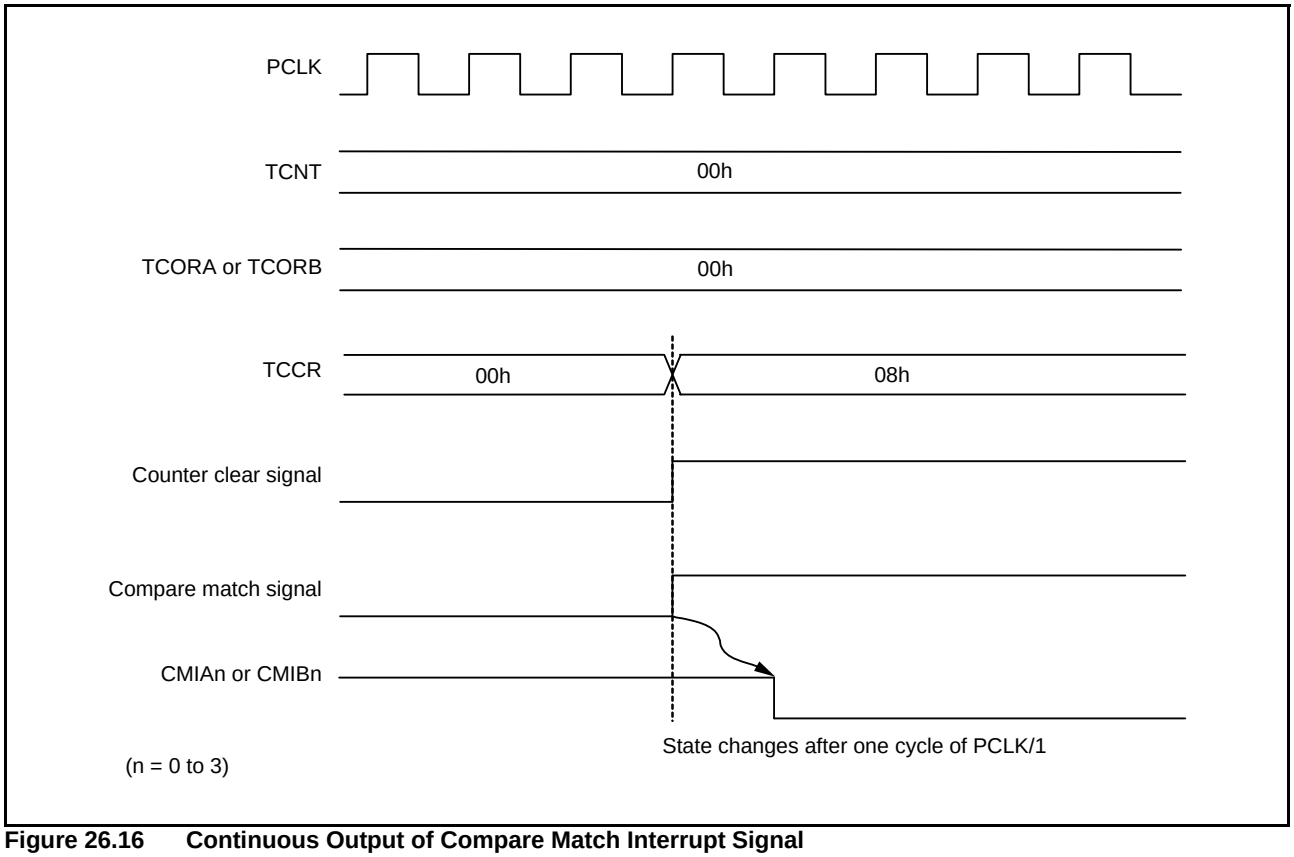


Figure 26.16 Continuous Output of Compare Match Interrupt Signal

27. Compare Match Timer (CMT)

The RX63N/RX631 Group 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.

27.1 Overview

Table 27.1 lists the specifications for the CMT.
Figure 27.1 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.

Table 27.1 Specifications of CMT

Item	Description
Count clock	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected individually for each channel.
Interrupt	A compare match interrupt can be requested individually for each channel.
Low power consumption facilities	Each unit can be placed in a module stop state.

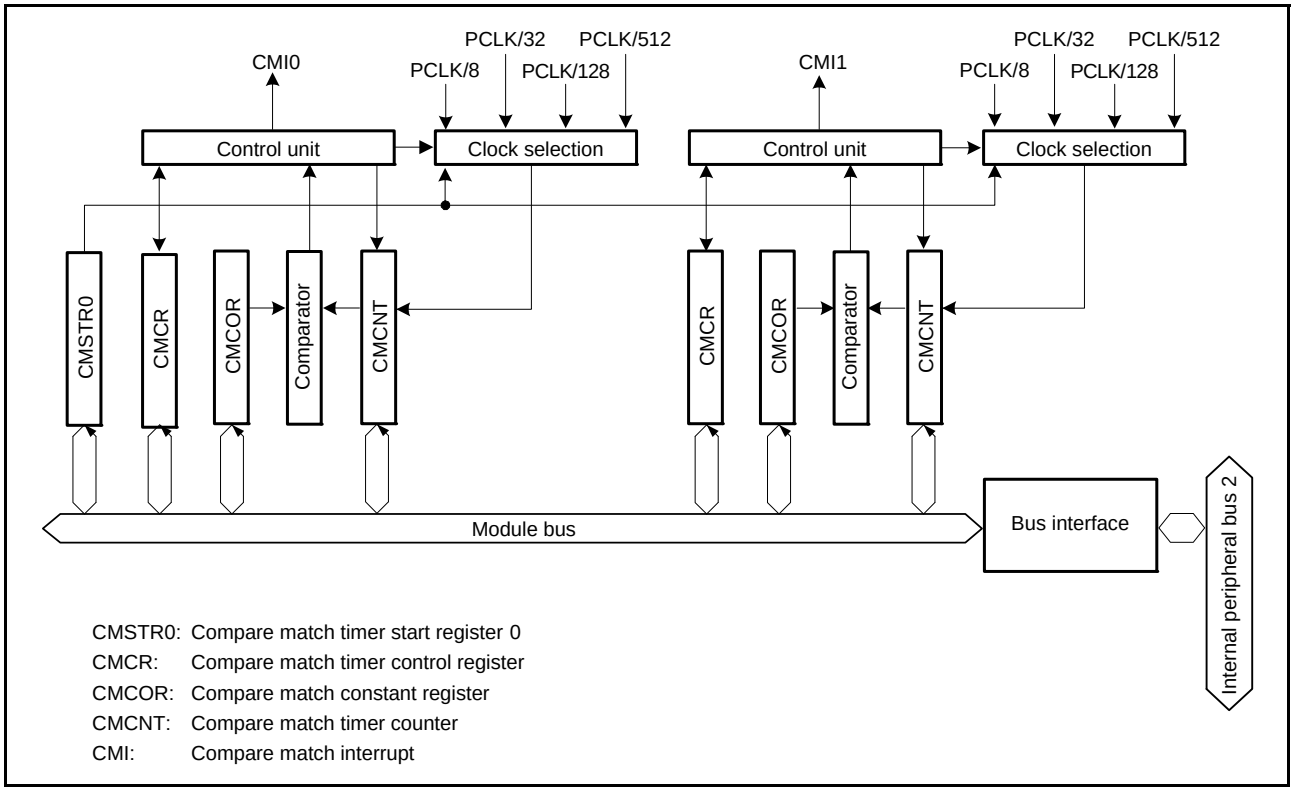


Figure 27.1 Block Diagram of CMT (Unit 0)

27.2 Register Descriptions

27.2.1 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

27.2.2 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

27.2.3 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 clock to be input to CMCNT from four frequency dividing clocks obtained by dividing the peripheral clock (PCLK).

When the STRn (n = 0 to 3) bit in CMSTRm (m = 0 or 1) is set to 1, CMCNT starts counting up on the clock selected with bits CKS[1:0].

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when CMCNT and CMCOR values match.

27.2.4 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

CMCNT is a readable/writable up-counter to generate interrupt requests.
When an frequency dividing clock is selected by bits CKS[1:0] in CMCR and the STRn (n = 0 to 3) bit in CMSTRm (m = 0 or 1) is set to 1, CMCNT starts counting up using the selected clock.
When the value in CMCNT and the value in CMCOR match, CMCNT is cleared to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

27.2.5 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

CMCOR is a readable/writable register to set a compare match cycle with CMCNT.

27.3 Operation

27.3.1 Periodic Count Operation

When an frequency dividing clock is selected by bits CKS[1:0] in CMCR and the STRn (n = 0 to 3) bit in CMSTRm (m = 0 or 1) is set to 1, CMCNT starts counting up using the selected clock.
When the value in CMCNT and the value in CMCOR match, CMCNT is cleared to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated. CMCNT then starts counting up again from 0000h. Figure 27.2 shows the operation of the CMCNT counter.

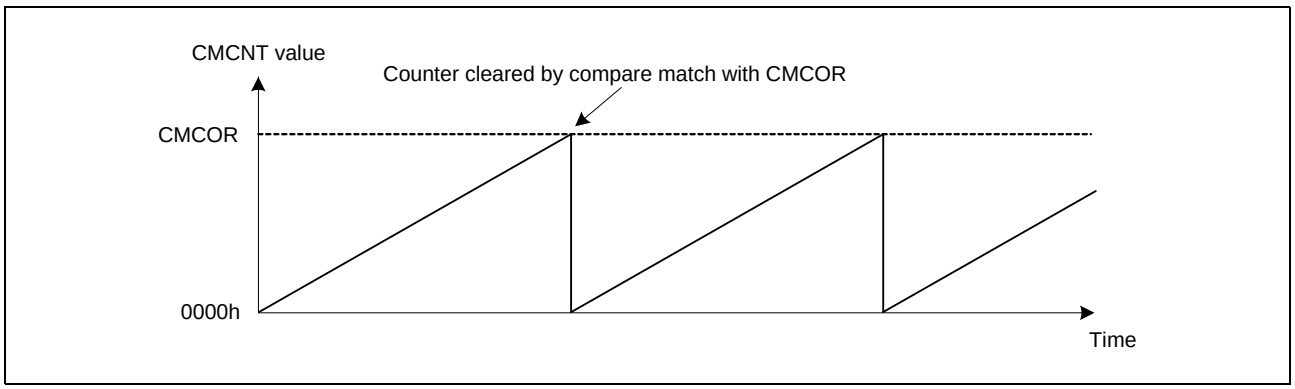


Figure 27.2 CMCNT Counter Operation

27.3.2 CMCNT Count Timing

As the count clock to be input to CMCNT, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral clock (PCLK) can be selected with the CKS[1:0] bits in CMCR. Figure 27.3 shows the timing of CMCNT.

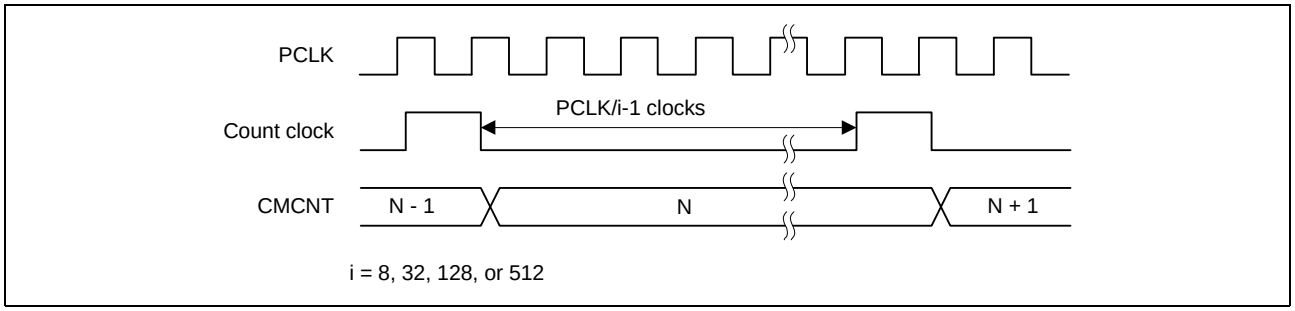


Figure 27.3 CMCNT Count Timing

27.4 Interrupts

27.4.1 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 is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 15, Interrupt Controller (ICUb).

Table 27.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation	DMAC Activation
CMI0	Compare match between CMT0.CMCNT and CMT0.CMCOR	Possible	Possible
CMI1	Compare match between CMT1.CMCNT and CMT1.CMCOR	Possible	Possible
CMI2	Compare match between CMT2.CMCNT and CMT2.CMCOR	Possible	Possible
CMI3	Compare match between CMT3.CMCNT and CMT3.CMCOR	Possible	Possible

27.4.2 Timing of Compare Match Interrupt Generation

When CMCNT and CMCOR 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 CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT input clock.
Figure 27.4 shows the timing of interrupt flag setting to 1.

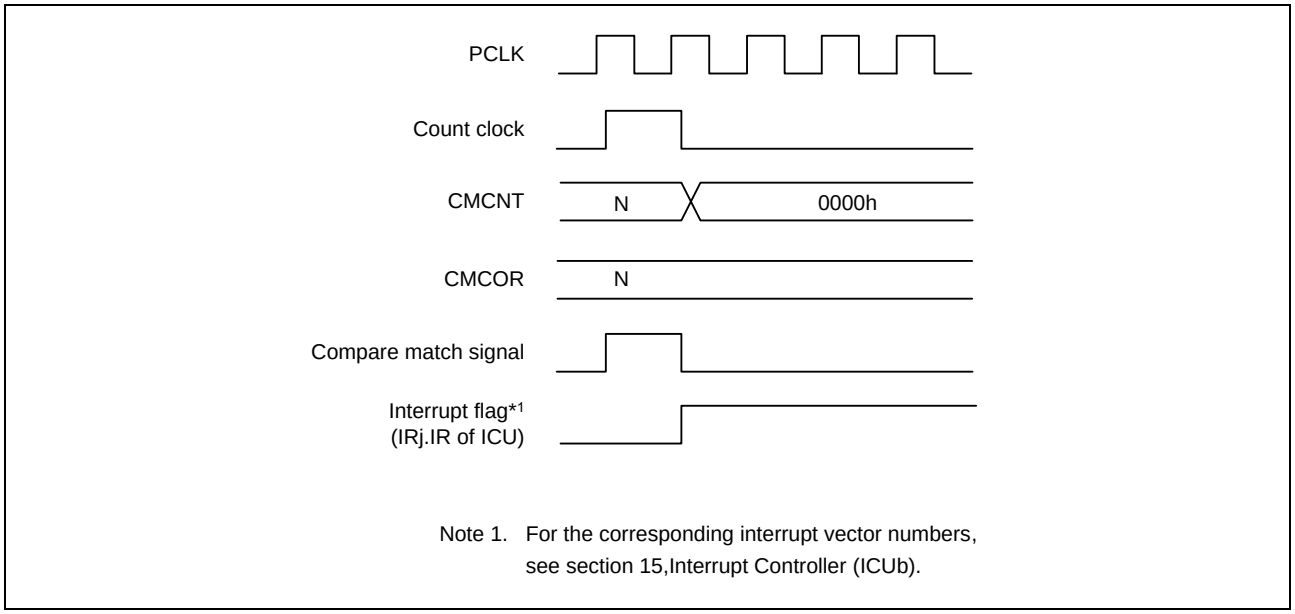


Figure 27.4 Timing of Interrupt Flag Setting to 1 at Compare Match

27.5 Usage Notes

27.5.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. The CMT is disabled by default. The registers can be accessed by canceling the module stop state. For details, see section 11, Low Power Consumption.

27.5.2 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 27.5 shows the timing to clear the CMCNT counter.

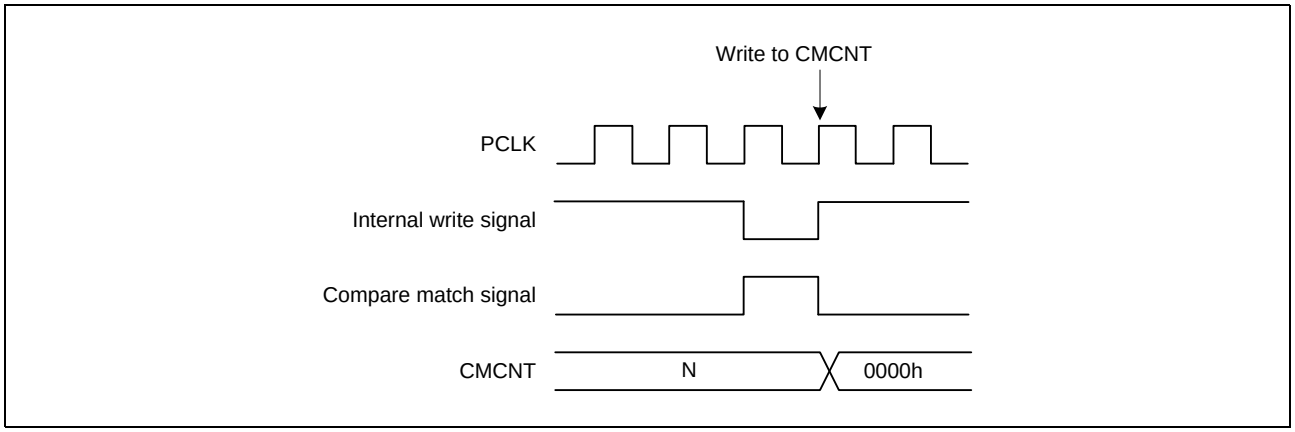


Figure 27.5 Conflict between Write and Compare Match Processes of CMCNT

27.5.3 Conflict between Write and Count-Up Processes of CMCNT

Even when the count-up occurs while writing to CMCNT, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 27.6 shows the timing to write the CMCNT counter.

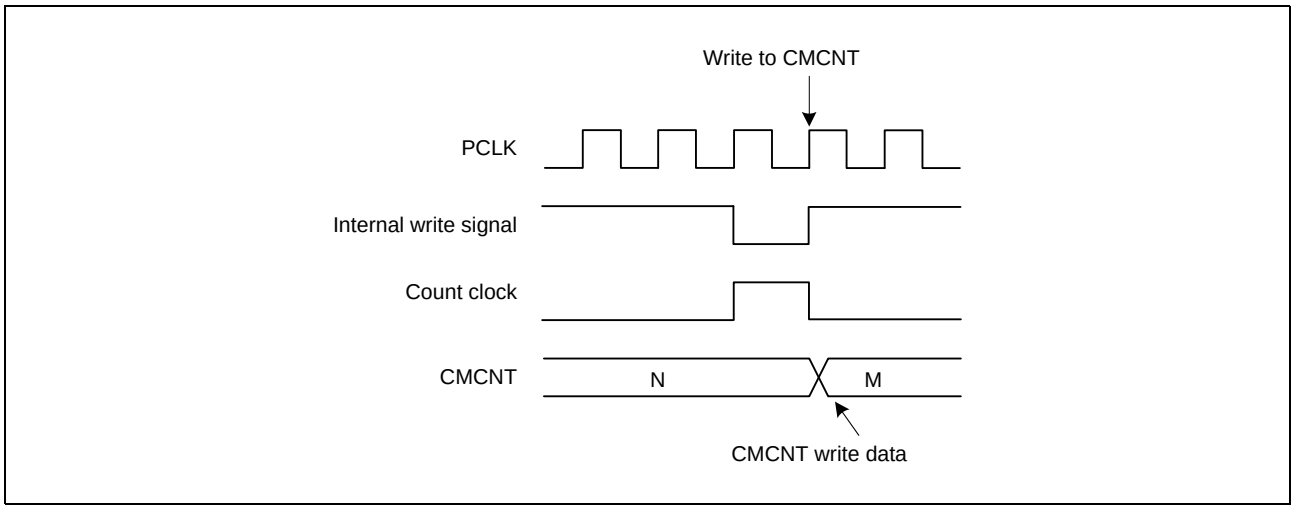


Figure 27.6 Conflict between Write and Count-Up Processes of CMCNT

28. Realtime Clock (RTCa)

28.1 Overview

The RTC is capable of counting 100 years from year 0 to year 99. If we place 20 in the hundreds and thousands positions, counting is automatically adjusted for leap years.
The source to drive counting of the time counters is selectable as the sub-clock or a main clock.
With the source for counting pre-scaled to produce a 128-Hz clock signal as the base clock, cycles of this signal are counted in year, month, date, day of the week, a.m. or p.m. (in 12-hour mode), hour, minute, second, and 1/128 second units.

Table 28.1 lists the specifications of the RTC, Figure 28.1 shows a block diagram of the RTC, and Table 28.2 shows the pin configuration of the RTC.

Table 28.1 Specifications of RTCA

Item	Description
Count source	Sub-clock (XCIN) or main clock (EXTAL)
Clock and calendar functions	• Year, month, date, day of the week, hours, minutes, and seconds are counted and represented in BCD • Indicates the state of 1, 2, 4, 8, 16, 32, or 64 Hz in binary • Selection of 12- or 24-hour mode • Start/stop function • 30-second adjustment (30 seconds or less are rounded down to 00 second, and 30 seconds or more are rounded up to one minute) • Automatic leap year adjustment • Output a 1-Hz clock • Time error adjustment function
Interrupts	• Alarm interrupt (ALM) Year, month, date, day of the week, hours, minutes, and seconds can be selected as conditions for the alarm interrupt • 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) Indicates occurrence of a carry to the seconds counter or a carry to the 64-Hz counter during reading of the 64-Hz counter • Recovery from software standby mode or deep software standby mode can be performed by an alarm interrupt or periodic interrupt
Time-capture facility	• Times when any of three event signals are input can be captured The month, date, hour, minute, and second are captured for each event

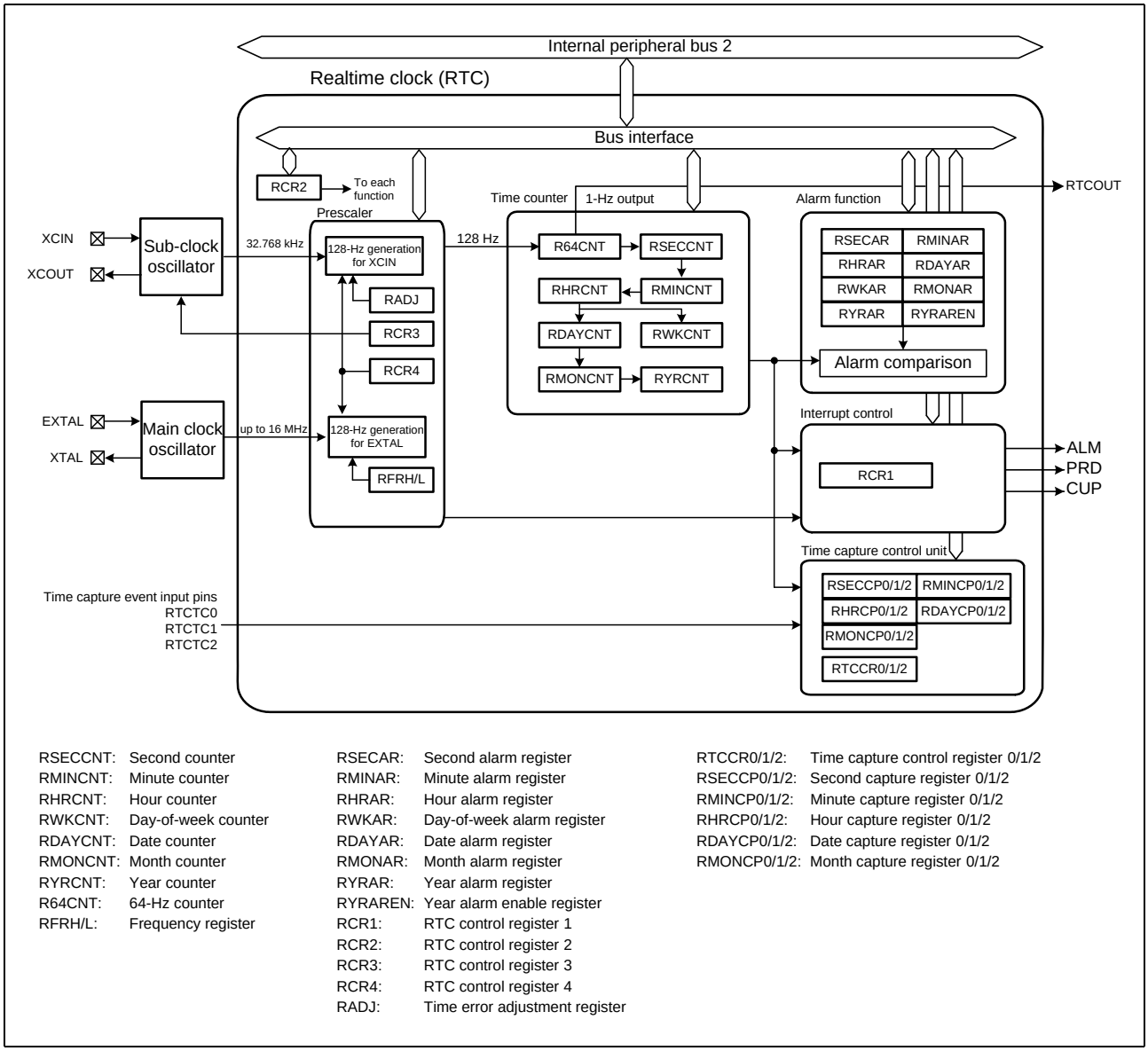


Figure 28.1 Block Diagram of RTC

Table 28.2 Pin Configuration of RTC

Pin Name	I/O	Function
XCIN	Input	Connect a 32.768-kHz crystal resonator for the RTC to these pins. The external clock can also be input to the XCIN pin.
XCOUT	Output	
EXTAL	Input	Connect a crystal resonator for the main clock. For details, see section 9, Clock Generation Circuit.
XTAL	Output	
RTCOUT	Output	Output a 1-Hz waveform, but this clock is not output in deep software standby mode.
RTCIC0	Input	Pins for time capture event input
RTCIC1	Input	
RTCIC2	Input	

28.2 Register Descriptions

Furthermore, when writing to or reading from RTC registers, do so in accord with section 28.5.5, Points for Caution 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 or in deep standby mode. 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, date, day of the week, hours, minutes, seconds, and 64-Hz counters continue to operate. However, 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 just after setting any of these registers. For details, refer to section, section 28.5.4, Transitions to Low Power Consumption Modes after Setting Registers.

28.2.1 64-Hz Counter (R64CNT)

Address(es): 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	64Hz	Indicate the state between 64 Hz and 1 Hz.	R
b1	F32HZ	32Hz		R
b2	F16HZ	16Hz		R
b3	F8HZ	8Hz		R
b4	F4HZ	4Hz		R
b5	F2HZ	2Hz		R
b6	F1HZ	1Hz		R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock.
The state in the sub-seconds range can be confirmed by reading this counter.
This counter is cleared to “00h” by a RTC software reset or executing 30-second adjustment.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

28.2.2 Second Counter (RSECCNT)

Address(es): 0008 C402h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SEC10[2:0]			SEC1[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	SEC1[3:0]	Ones Place of Seconds	Counts from 0 to 9 once per second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	Tens Place of Seconds	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RSECCNT 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.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

28.2.3 Minute Counter (RMINCNT)

Address(es): 0008 C404h

	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]	Ones Place of Minutes	Counts from 0 to 9 once per minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	Tens Place of Minutes	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RMINCNT 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 (practically 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.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

28.2.4 Hour Counter (RHRCNT)

Address(es): 0008 C406h

	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]	Ones Place of Hours	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]	Tens Place of Hours	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 0.	R/W

RHRCNT 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 time mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (practically in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (practically 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.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

28.2.5 Day-of-Week Counter (RWKCNT)

Address(es): 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 0.	R/W

RWKCNT is used for setting and counting in the BCD-coded day-of-week value. It counts carries generated once per day in the hour counter.

A value from 0 through 6 (practically 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.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

28.2.6 Date Counter (RDAYCNT)

Address(es): 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]	Ones Place of Days	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]	Tens Place of Days	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

RDAYCNT 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 (practically 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.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

28.2.7 Month Counter (RMONCNT)

Address(es): 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]	Ones Place of Months	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	Tens Place of Months	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

RMONCNT 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 (practically 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.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

28.2.8 Year Counter (RYRCNT)

Address(es): 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]	Ones Place of Years	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]	Tens Place of Years	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

RYRCNT 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 (practically 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.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

28.2.9 Second Alarm Register (RSECAR)

Address(es): 0008 C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	SEC10[2:0]			SEC1[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	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	When this bit is set to 1, this register value is compared with the RSECCNT value.	R/W

RSECAR 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 corresponding IR92.IR flag of the ICU is set to 1.
RSECAR values from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
RSECAR is cleared to 00h when the RTC software reset bit (RCR2.RESET) is set to 1.

28.2.10 Minute Alarm Register (RMINAR)

Address(es): 0008 C412h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	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	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	When this bit is set to 1, this register value is compared with the RMINCNT value.	R/W

RMINAR 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 corresponding IR92.IR flag of the ICU is set to 1.

RMINAR values from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RMINAR is cleared to 00h when the RTC software reset bit (RCR2.RESET) is set to 1.

28.2.11 Hour Alarm Register (RHRAR)

Address(es): 0008 C414h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	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	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	PMBit	Time Alarm Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RHRCNT value.	R/W

RHRAR 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 corresponding IR92.IR flag of the ICU is set to 1.
The specifiable time differs according to the setting in the time mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (practically in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (practically 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.
RHRAR is cleared to 00h when the RTC software reset bit (RCR2.RESET) is set to 1.

28.2.12 Day-of-Week Alarm Register (RWKAR)

Address(es): 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 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RWKCNT value.	R/W

RWKAR is an alarm register corresponding to the BCD-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 corresponding IR92.IR flag of the ICU is set to 1.

RWKAR values from 0 through 6 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RWKAR is cleared to 00h when the RTC software reset bit (RCR2.RESET) is set to 1.

28.2.13 Date Alarm register (RDAYAR)

Address(es): 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 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RDAYCNT value.	R/W

RDAYAR 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 corresponding IR92.IR flag of the ICU is set to 1.

RDAYAR values from 01 through 31 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RDAYAR is cleared to 00h when the RTC software reset bit (RCR2.RESET) is set to 1.

28.2.14 Month Alarm Register (RMONAR)

Address(es): 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 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RMONCNT value.	R/W

RMONAR 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 corresponding IR92.IR flag of the ICU is set to 1.

RMONAR values from 01 through 12 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RMONAR is cleared to 00h when the RTC software reset bit (RCR2.RESET) is set to 1.

28.2.15 Year Alarm Register (RYRAR)

Address(es): 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

RYRAR is an alarm register corresponding to the BCD-coded year counter RYRCNT.
RYRAR values from 00 through 99 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
RYRAR is cleared to 0000h when the RTC software reset bit (RCR2.RESET) is set to 1.

28.2.16 Year Alarm Enable Register (RYRAREN)

Address(es): 0008 C41Eh

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, the RYRAR value is compared with the RYRCNT value.	R/W

When the ENB bit in RYRAREN 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 corresponding IR92.IR flag of the ICU is set to 1.
RYRAREN is cleared to 00h when the RTC software reset bit (RCR2.RESET) is set to 1.

28.2.17 RTC Control Register 1 (RCR1)

Address(es): 0008 C422h

	b7	b6	b5	b4	b3	b2	b1	b0
	PES[3:0]				—	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	—	Reserved	This bit is read as 0. The write value should be 0.	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. (However, when the main clock is selected (RCR4.RCKSEL = 1) while PES[3:0] = 0110b, a periodic interrupt is generated every 1/128 second.) 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

Bits AIE, PIE, and PES[3:0] are updated in synchronization with the count source. When RCR1 is modified, check that all the bits have been updated without fail before continuing with further 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 mode is canceled regardless of the value of the AIE bit.

CIE Bit (Carry Interrupt Enable)

This bit enables or disables interrupt requests when a carry to the second counter occurs or a carry to the 64-Hz counter occurs during read access to the 64-Hz counter.

PIE Bit (Periodic Interrupt Enable)

This bit enables or disables a periodic interrupt.
If the times indicated by the counters and PES[3:0] settings match in deep software standby mode, the mode is canceled regardless of the value of the PIE bit.

PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt (PRD) is generated with the period specified by these bits.

28.2.18 RTC Control Register 2 (RCR2)

Address(es): 0008 C424h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	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: Year, month, day-of-week, date, hour, minute, second, and 64-Hz counters, and prescaler are stopped. 1: Year, month, day-of-week, date, hour, minute, second, and 64-Hz counters, and prescaler operate normally.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Writing is invalid. 1: The prescaler and target registers are reset by RTC software reset. • In reading 0: In normal clock operation, or RTC software reset has completed. 1: During a RTC software reset	R/W
b2	ADJ30	30-Second Adjustment	• In writing 0: Writing is invalid. 1: 30-second adjustment is executed. • In reading 0: In normal clock operation, or 30-second adjustment has completed. 1: During 30-second adjustment	R/W
b3	RTCOE	RTCOUT Output Enable	0: Clock signals are not output from the RTCOUT pin. 1: Clock signals are output from the RTCOUT pin.	R/W
b4	AADJE	Automatic Adjustment Enable*1	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*1	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every one minute. 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every ten seconds.	R/W
b6	HR24	Hours Mode	0: The RTC operates in 12-hour mode. 1: The RTC operates in 24-hour mode.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

Note 1. When the main clock is selected, the AADJE and AADJP settings are invalid.

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 without fail, and then make next settings.

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 cleared to 0.

When 1 is written to the RESET bit, check that the bit is cleared 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 one 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 cleared to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is cleared 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 cleared to 0 by RTC software reset.

RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-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 a 1-Hz clock signal from the RTCOUT pin is to be output to an external pin, enable the port control as well as setting this bit.

AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (PMADJ[1:0] in RADJ) to 00h (no adjustment) before changing the value of the AADJE bit.
The AADJE bit is cleared to 0 by RTC software reset.

AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Set the plus–minus bits (PMADJ[1:0] in RADJ) to 00h (no adjustment) before changing the value of the AADJP bit.
The AADJP bit is cleared to 0 by 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.

28.2.19 RTC Control Register 3 (RCR3)

Address(es): 0008 C426h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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 Control	0: Clock input from the sub-clock oscillator is disabled. 1: Clock input from the sub-clock oscillator is enabled.	R/W
b3 to b1	—	Reserved	Set these bits to 110b.* They are read as 110b.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewriting bits b3 to b1 while the sub-clock oscillator is running is prohibited. Rewrite bits b3 to b1 only when the sub-clock oscillator stop bit (SOSCCR.SOSTP) of the sub-clock oscillator control register is set to 1 and the sub-clock oscillator control bit (RCR3.RTCEN) of the RTC control register 3 is set to 0.

The RCR3 register is used to control the sub-clock oscillator in the clock generation circuit. See section 9., Clock Generation Circuit, for the control of the sub-clock oscillator.
Once rewriting the RCR3 register, do not perform the next processing until you make sure that all bit values are updated.

RTCEN Bit (Sub-Clock Generator Control)

This bit controls the start/stop operation of the sub-clock generator. Start/stop of the sub-clock oscillator is controlled by the RTCEN bit and registers in the clock generation circuit. The sub-clock oscillator is made active when either bit is set to "start,"
When using the count source for the RTCA, use the RTCEN bit to make the start/stop operation setting for the sub-clock oscillator.
If the extended clock is selected (RCR4.RCKSEL bit = 1), the start/stop operation of the sub-clock oscillator is controlled by registers in the clock generation circuit regardless of the value of the RTCEN bit.

28.2.20 RTC Control Register 4 (RCR4)

Address(es): 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: The sub-clock is selected. 1: The main clock is selected.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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.

28.2.21 Frequency Register H/L (RFRH/L)

• RFRH register

Address(es): 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

• RFRL register

Address(es): 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 Comparison Value	To produce the operation clock from the main clock, this bit sets the comparison value of the 128-Hz clock cycle.	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 Comparison Value	To prepare the operation clock from the main clock, these bits set the comparison value of the 128-Hz clock cycle.	R/W

RFRH/L is a register to control the prescaler when the main clock is selected.
The RTC time counter operates on a 128-Hz clock signal as the base clock. Therefore, when the main clock is selected, the main clock is divided by the prescaler to produce a 128-Hz clock signal. Set the frequency comparison value in bits RFC[16:0] to produce a 128-Hz clock from the main clock frequency. The frequency comparison value should be changed depending on the main clock frequency. As for the calculation method, refer to the formula below.
A value from 0000 0007h through 0001 FFFFh can be specified as the frequency comparison value; 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.
In addition, when the main clock is 16.778 MHz or more, the frequency comparison value cannot be set.
The operating frequency of the peripheral module clock and the main clock should be in the relationship that peripheral module clock ≥ main clock.

- Calculation method of frequency comparison value

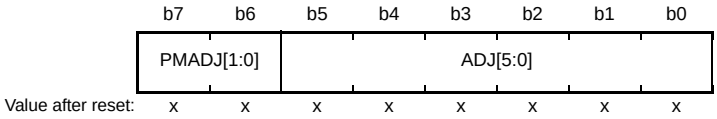
RFC[16:0] = (Main clock frequency)/128 – 1

Table 28.3 RFRH/L Register Settings by the Main Clock Frequency

Main Clock Frequency	RFRH/L Register Settings
4 MHz	0000 7A11h
8 MHz	0000 F423h
10 MHz	0001 312Ch
12 MHz	0001 6E35h
16 MHz	0001 E847h

28.2.22 Time Error Adjustment Register (RADJ)

Address(es): 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

RADJ is a register to adjust the time error.
Adjustment is performed by the addition to or subtraction from the prescaler.
When the sub-clock is selected, settings in RADJ are valid.
When the main clock is selected, adjustment is not performed regardless of settings in RADJ.
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 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 without fail before continuing with further processing.
This register is cleared to 00h by RTC software reset.

ADJ[5:0] Bits (Adjustment Value)

These bits specify the adjustment value (clock cycle number of sub-clock) 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.

28.2.23 Time Capture Control Register y (RTCCRy) (y = 0 to 2)

Address(es): RTCCR0 0008 C440h, RTCCR1 0008 C442h, RTCCR2 0008 C444h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCEN	—	TCNF[1:0]		—	TCST	TCCT[1:0]	
Value after reset:	x	0	x	x	0	x	x	x

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	Indicates that an event has been detected. Writing 0 clears this bit. 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 of 32-division).	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 invalid as the time capture event input. 1: The RTCICn pin is valid as the time capture event input. (n = 0 to 2)	R/W

Note 1. Writing 1 to this bit has no effect.

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 without fail before continuing with further processing.
This register is cleared to 00h by RTC software reset.

TCCT[1:0] Bits (Time Capture Control)

These pins 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 Star)

This pin 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 during the count operation is stopped (the RCR2.START bit is 0), the captured value is not guaranteed. In this case, write 0 to clear the TCST bit for deleting the captured value.
Writing 0 clears the TCST bit. In addition, writing any other value except 0 has no effect.
Clear the TCST bit while the TCCT[1:0] bits are 00b (no event is detected).
The TCST bit is cleared in synchronization with the count source. When the TCST bit is cleared, check that the bit has been updated without fail before continuing with further processing.

TCNF[1:0] Bits (Time Capture Noise Filter Control)

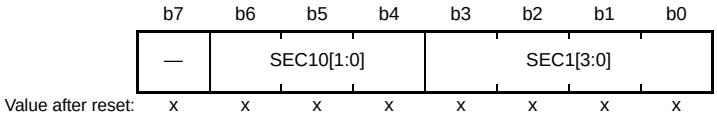
These bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).
When the noise filter is on, the count source of 1- and 32-division is selectable. In this case, when the input level on the time capture event input pin matches three times at the set sampling period, 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 functions of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) are multiplexed, enable the port control as well as setting this bit. In this case, the port control should be enabled first. If the TCEN bit is set to 0, set also the TCCT[1:0] bits to 00h.

28.2.24 Second Capture Register y (RSECCPy) (y = 0 to 2)

Address(es): RSECCP0 0008 C452h, RSECCP1 0008 C462h, RSECCP2 0008 C472h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	Ones Place of Seconds Captured	Capture value for the ones place of seconds	R
b6 to b4	SEC10[1:0]	Tens Place of Seconds Captured	Capture value for the tens place of seconds	R
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RSECCPy is a readable 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 cleared to 00h by 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.

28.2.25 Minute Capture Register y (RMINCPy) (y = 0 to 2)

Address(es): RMINCP0 0008 C454h, RMINCP1 0008 C464h, 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]	Ones Place of Minutes Captured	Capture value for the ones place of minutes	R
b6 to b4	MIN10[2:0]	Tens Place of Minutes Captured	Capture value for the tens place of minutes	R
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RMINCPy is a readable 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 cleared to 00h by 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.

28.2.26 Hour Capture Register y (RHRCPy) (y = 0 to 2)

Address(es): RHRCP0 0008 C456h, RHRCP1 0008 C466h, 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]	Ones Place of Hours Captured	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	Tens Place of Hours Captured	Capture value for the tens place of hours	R
b6	PM	PM	Time Captured Value for a.m./p.m. 0: a.m. 1: p.m.	R
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RHRCPy is a readable 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 cleared to 00h by 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.

28.2.27 Date Capture Register y (RDAYCPy) (y = 0 to 2)

Address(es): RDAYCP0 0008 C45Ah, RDAYCP1 0008 C46Ah, RDAYCP2 0008 C47Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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]	Ones Place of Days Captured	Capture value for the ones place of days	R
b5, b4	DATE10[1:0]	Tens Place of Days Captured	Capture value for the tens place of days	R
b7, b6	—	Reserved	Set these bits to 0. They are read as 0.	R/W

RDAYCPy is a readable register that captures the RDAYCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCI2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.
This register is cleared to 00h by 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.

28.2.28 Month Capture Register y (RMONCPy) (y = 0 to 2)

Address(es): RMONCP0 0008 C45Ch, RMONCP1 0008 C46Ch, RMONCP2 0008 C47Ch

	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]	Ones Place of Months Captured	Capture value for the ones place of months	R
b4	MON10	Tens Place of Months Captured	Capture value for the tens place of months	R
b7 to b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

RMONCPy is a readable 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 cleared to 00h by 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.

28.3 Operation

28.3.1 Outline of Initial Settings of Registers after Power-On

After the power is turned on, the initial settings for the clock setting, time-error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

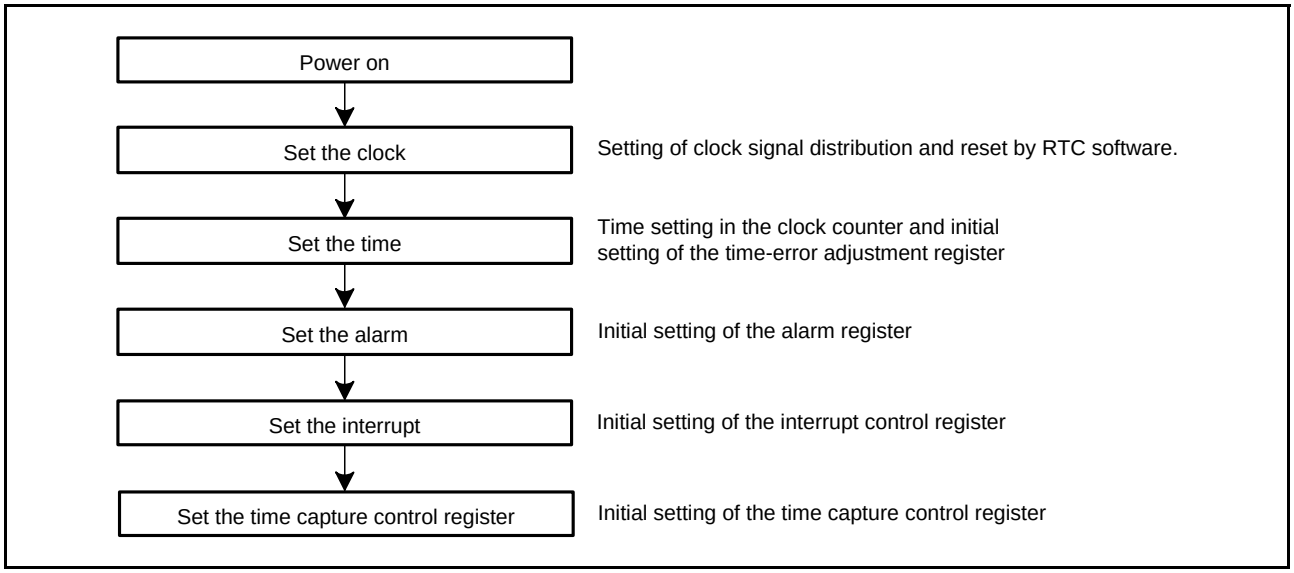


Figure 28.2 Outline of Initial Settings after Power-On

28.3.2 Clock Setting Procedure

Figure 28.3 shows how to set the clock.

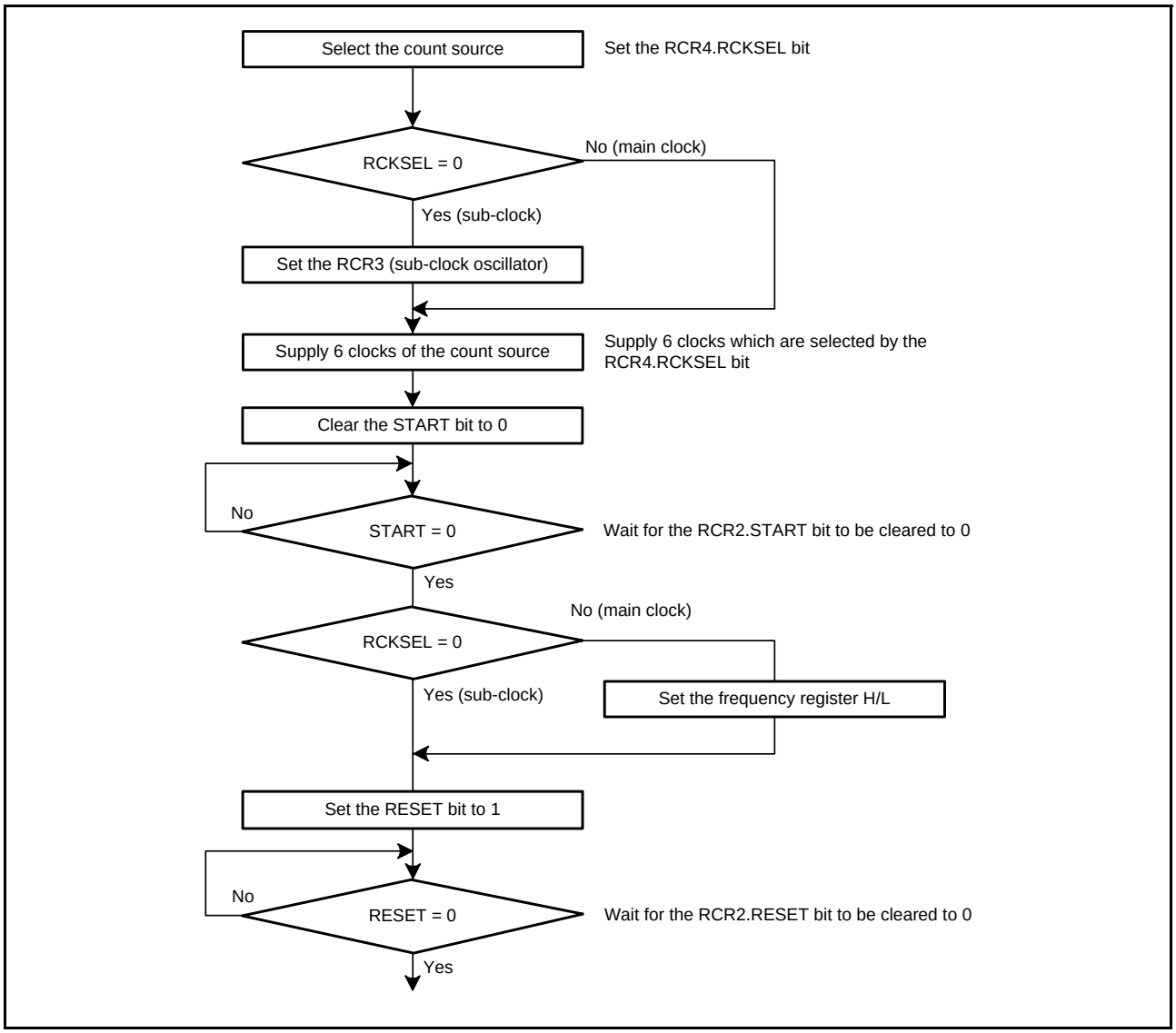


Figure 28.3 Clock Setting Procedure

28.3.3 Setting the Time

Figure 28.4 shows how to set the time.

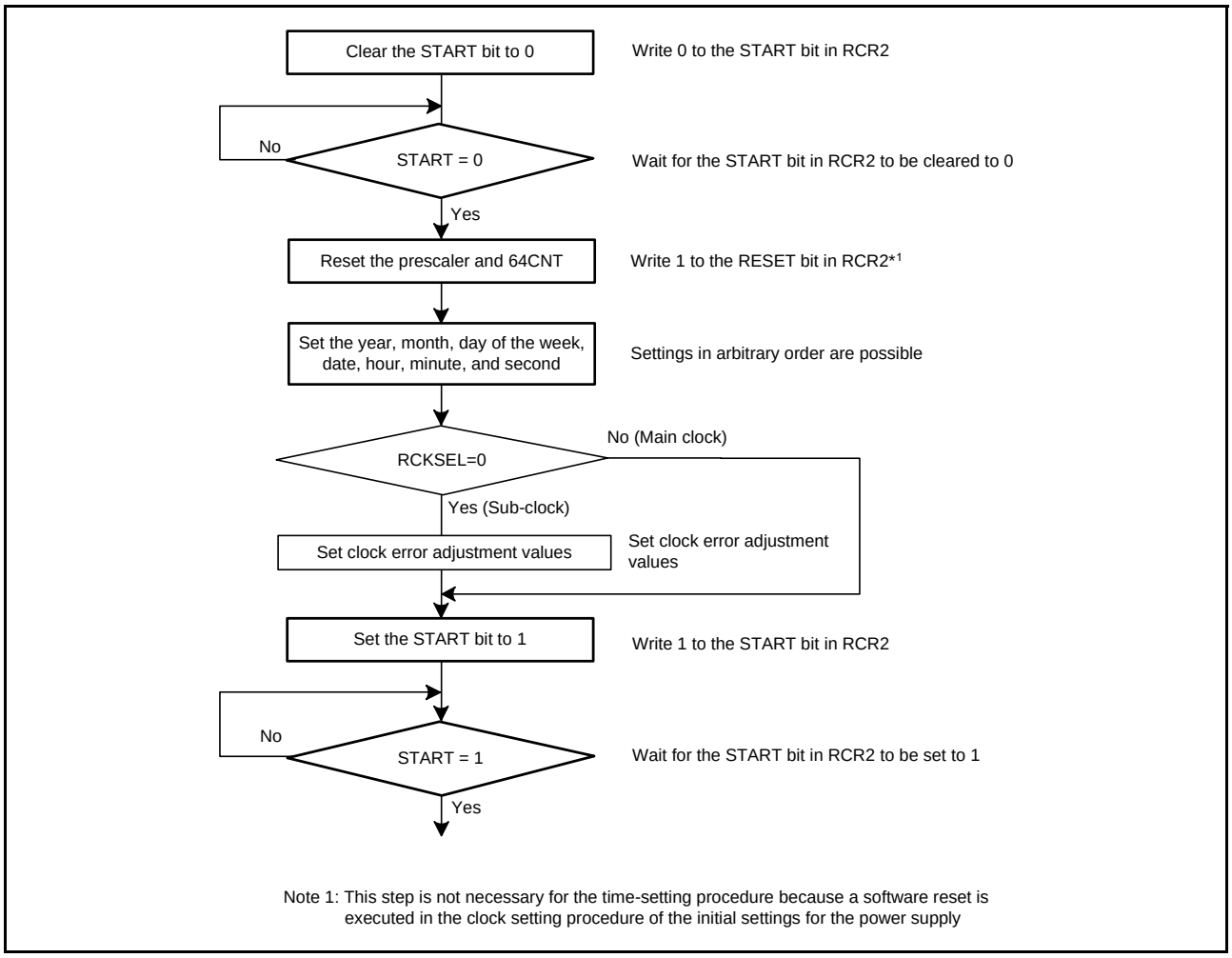


Figure 28.4 Setting the Time

28.3.4 30-Second Adjustment

Figure 28.5 shows how to execute 30-second adjustment.

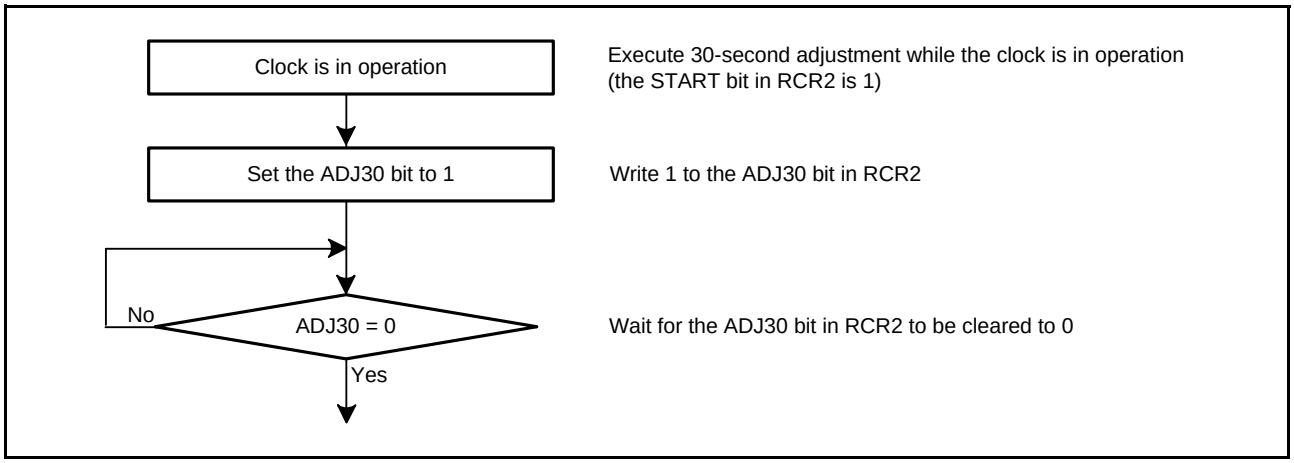


Figure 28.5 30-Second Adjustment

28.3.5 Reading 64-Hz Counter and Time

Figure 28.6 shows how to read the 64-Hz counter and time.

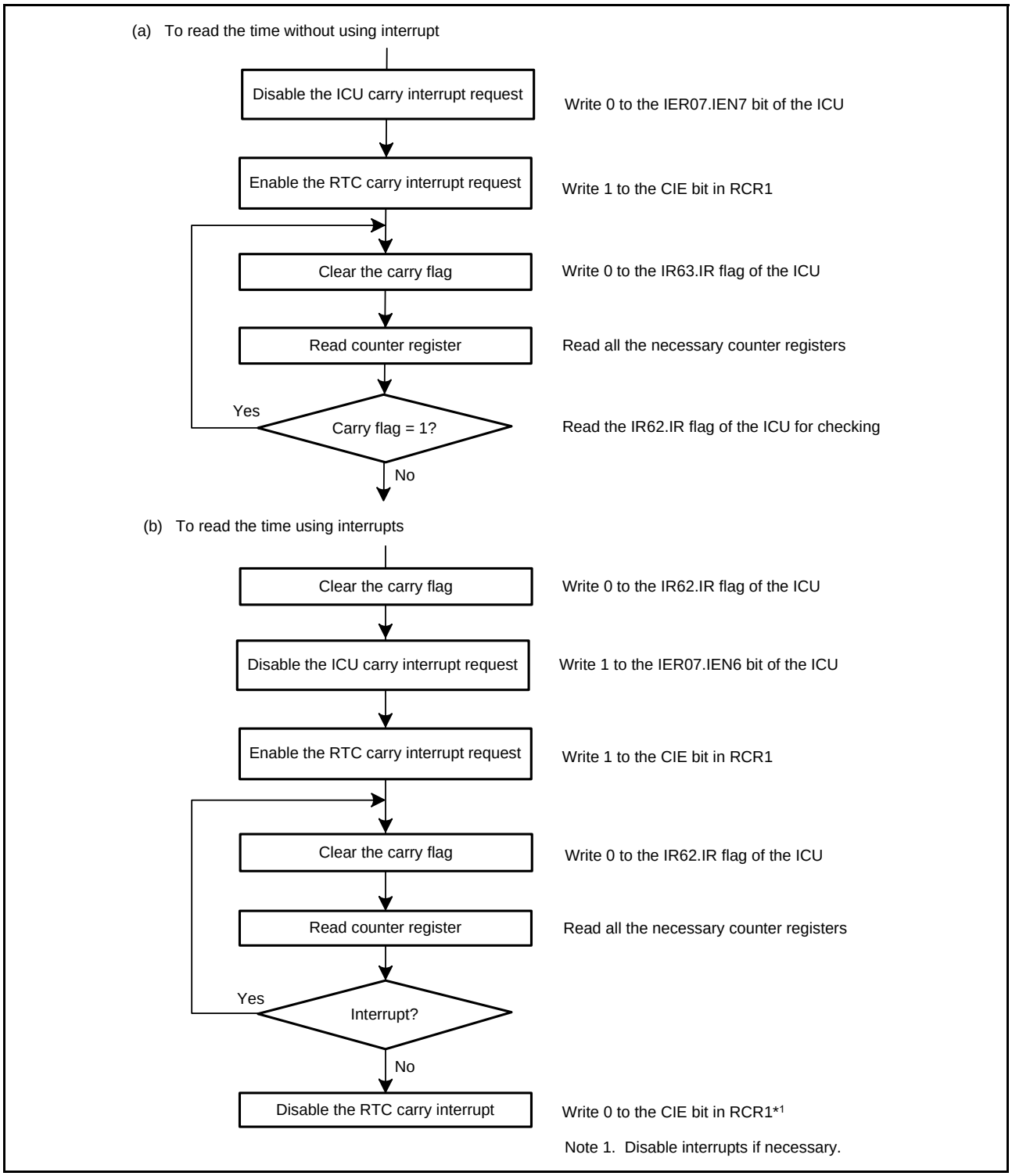


Figure 28.6 Reading Time

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), and the procedure using carry interrupts in (b) in Figure 28.6. To keep the program simple, method (a) should be used in most cases.

28.3.6 Alarm Function

Figure 28.7 shows how to use the alarm function.

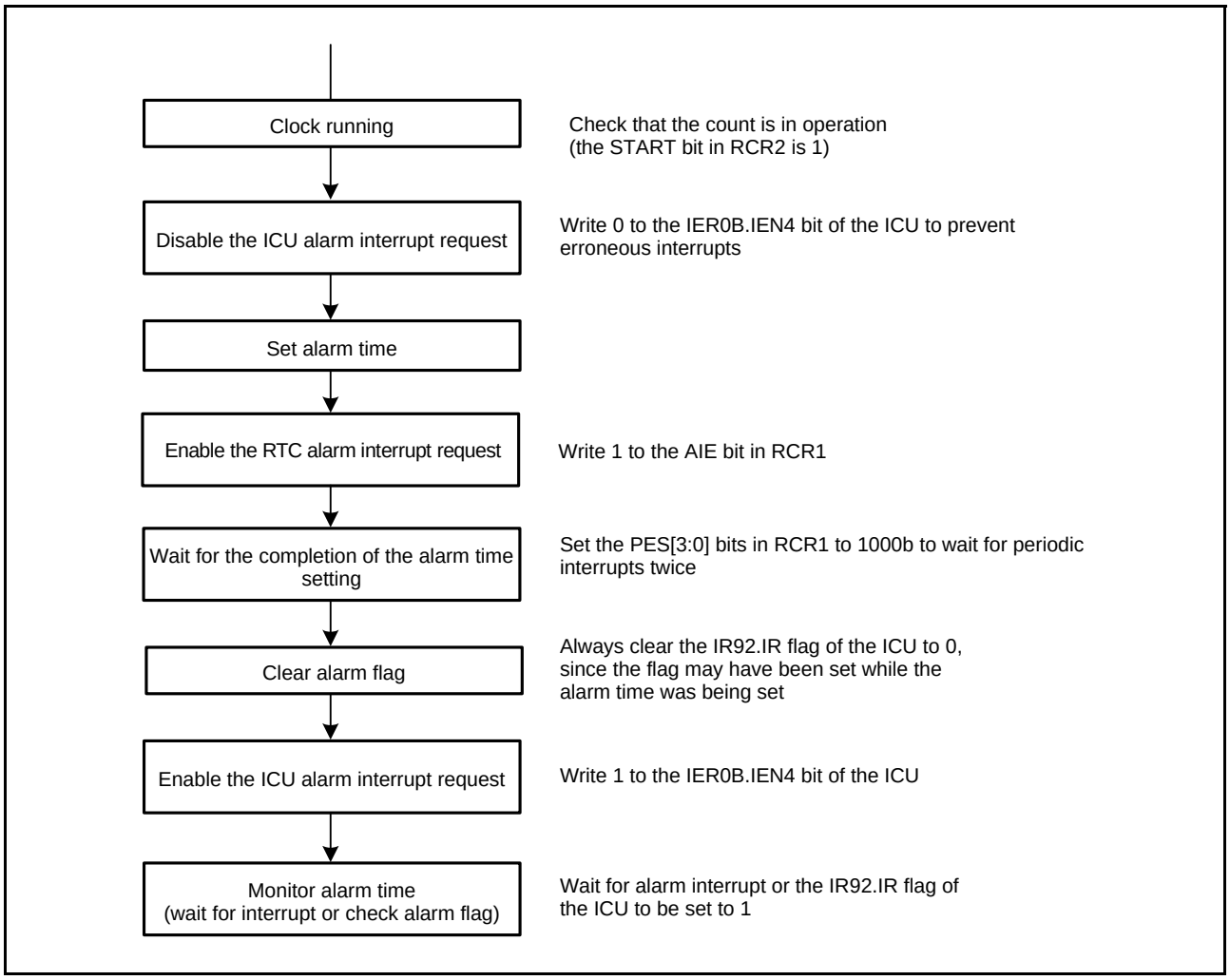


Figure 28.7 Using Alarm Function

An alarm can be generated by the year, month, date, day-of-week, hour, minute, or second value, or a combination of these. 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.

When the counter and the alarm time match, the IR92.IR flag of the ICU 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 IER0B.IEN4 bit of the ICU, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 clears the IR92.IR flag of the ICU.

When the counter and the alarm time match in a low power consumption state, this LSI returns from the low power consumption state.

28.3.7 Procedure for Disabling Alarm Interrupt

Figure 28.8 shows the procedure for disabling the enabled alarm interrupt request.

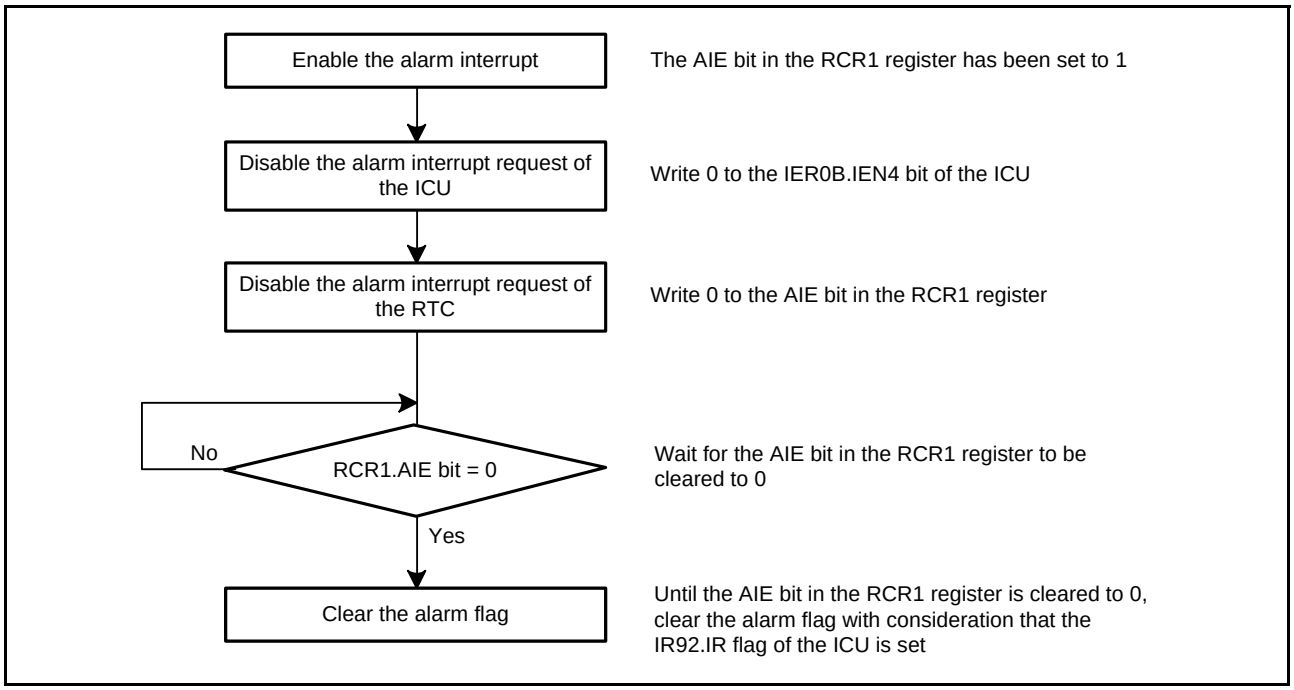


Figure 28.8 Procedure for Disabling Alarm Interrupt Request

28.3.8 Time Error Adjustment Function

The time-error correction 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 one second of operation, 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.

28.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the AADJE bit in RCR2 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 AADJP bit in RCR2 elapses.
Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
At 32.769 kHz, 1 second elapses on counting up by 1 from 32,768. Since adding 60 is suitable if we have 1 minute, the time can be adjusted by adding 60 to the value counted by the prescaler once per minute.

Register settings:

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (addition)
- RADJ. ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz
Adjustment procedure:
At 32.766 kHz, 1 second elapses on counting up by 2 from 32,768. Since adding 20 is suitable if we have 10 seconds, the time can be adjusted by adding 20 to the value counted by the prescaler once every 10 seconds.

Register settings:

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 10b (addition)
- RADJ. ADJ[5:0] = 20 (14h)

28.3.8.2 Adjustment by Software

Enable adjustment by software by setting the AADJE bit in RCR2 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:
At 32.769 kHz, 1 second elapses on counting up by 1 from 32,768. The time can be adjusted by adding 1 to the value counted by the prescaler once per second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (addition)
- RADJ. ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

28.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after clearing the PMADJ[1:0] bits in RADJ to 00h (no adjustment).

Changing from adjustment by software to automatic adjustment:
(1) Clear the PMADJ[1:0] bits in RADJ to 00h (no adjustment).
(2) Set the AADJE bit in RCR2 to 1 (enabling automatic adjustment).
(3) Use the AADJP bit in RCR2 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) Clear the PMADJ[1:0] bits in RADJ to 00h (no adjustment).
(2) Set the AADJE bit in RCR2 to 0 (enabling adjustment by software).
(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 RADJ.

28.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by clearing the PMADJ[1:0] bits in RADJ to 00h (no adjustment).

28.3.8.5 Capturing the Time

The RTC is capable of storing the month, date, hour, minute and second 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. When the level on the pin matches three times at the set sampling period, the noise filter conveys the level to the RTC's internal circuits, and this level is retained within the RTC until the level on the target pin for sampling has again matched 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 28.9 and operation when the noise filter is on is shown in Figure 28.10.

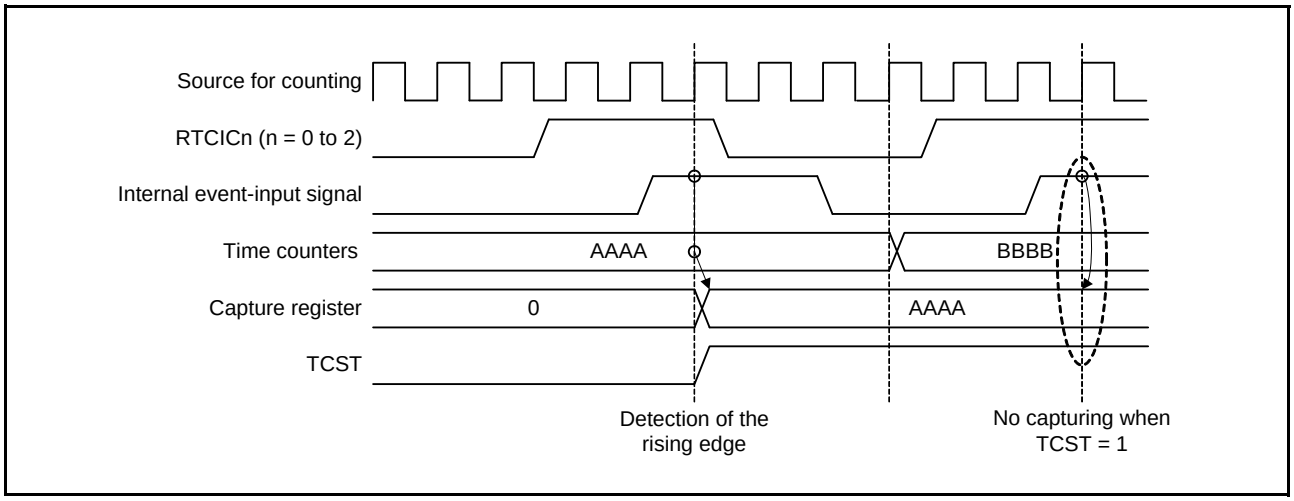


Figure 28.9 Timing of a Time-Capture Operation (with the Filter Off)

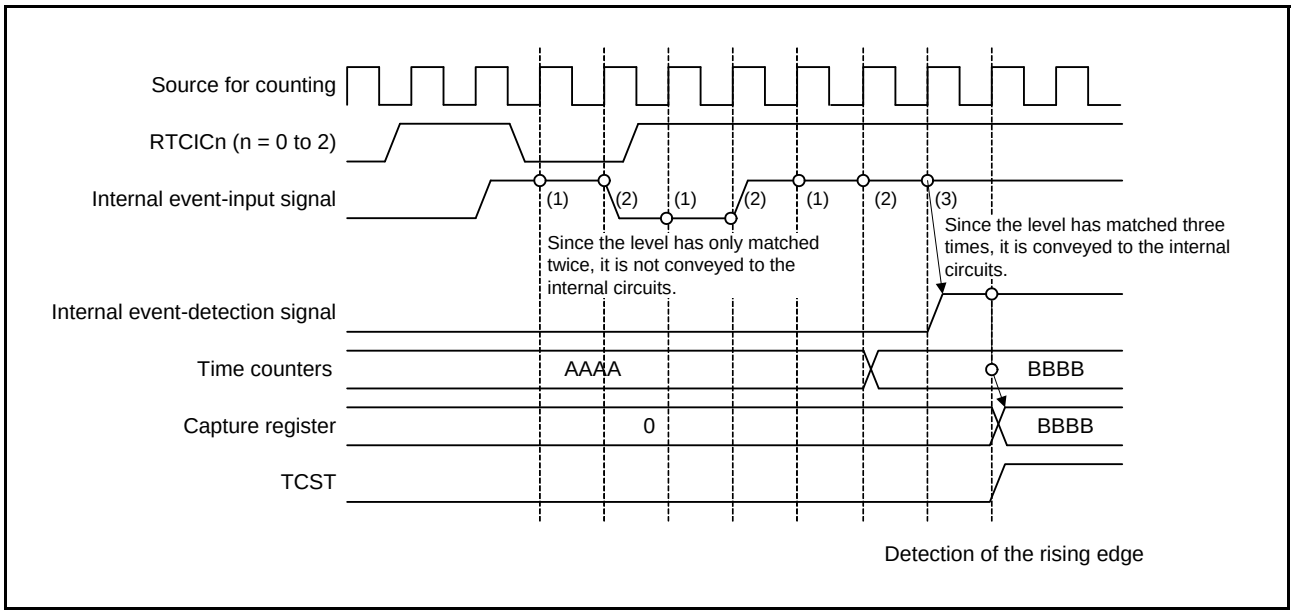


Figure 28.10 Timing of a Time-Capture Operation (with the Filter On)

28.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 28.4 lists interrupt sources for the RTC.

Table 28.4 RTC Interrupt Sources

Name	Interrupt Sources	Interrupt Status Flag*1
ALM	Alarm interrupt	IPR092.IR
PRD	Periodic interrupt	IPR093.IR
CUP	Carry interrupt	IPR062.IR

Note 1. For details, see section 15., Interrupt Controller (ICUb). To identify the interrupt source after return from the deep software standby mode, see the flags on the low power consumption function side. See section 11, Low Power Consumption Function, for details.

(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 the description of each alarm register).
Since there is a possibility of the interrupt flag being set when the settings of the alarm registers match the clock counters, clear the IR flag in IR92 for the interrupt after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been cleared 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.

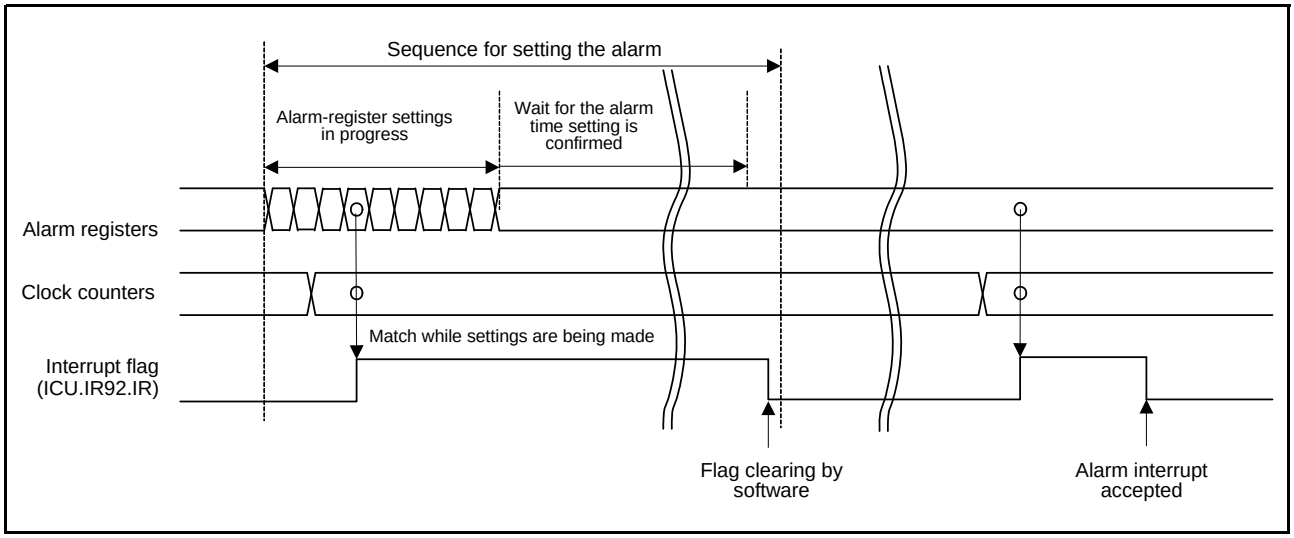


Figure 28.11 Timing Chart for the Alarm Interrupt (ALM)

(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 RCR1.PES[3:0] bits.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter occurs or a carry to the R64CNT counter occurs during read access to the 64-Hz counter.

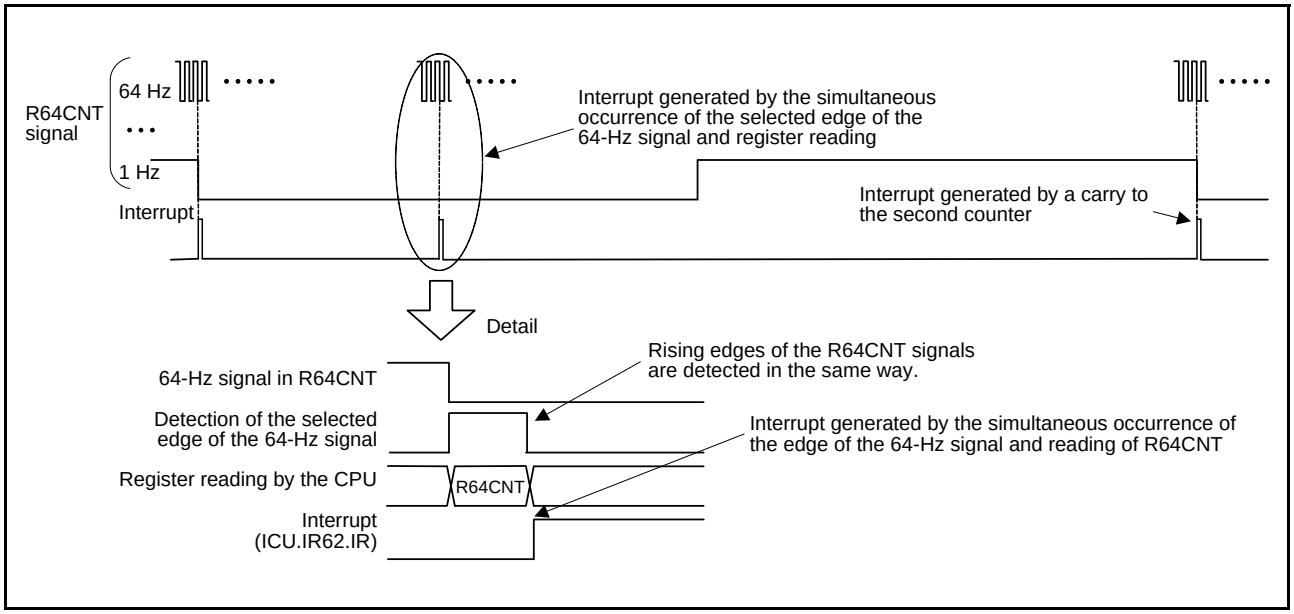


Figure 28.12 Carry Interrupt (CUP) Timing Chart

28.5 Usage Notes

28.5.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT,
RCR2.RTCOE, RCR2.HR24, RFRH, RFRL

The counter must be stopped before writing to any of the above registers.

28.5.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 28.13.
The generation and period of the periodic interrupt can be changed by the setting of the PES[3:0] bits in RCR1. However, since the prescaler, R64CNT, and RSECCNT are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the PES[3:0] bits in RCR1.
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.

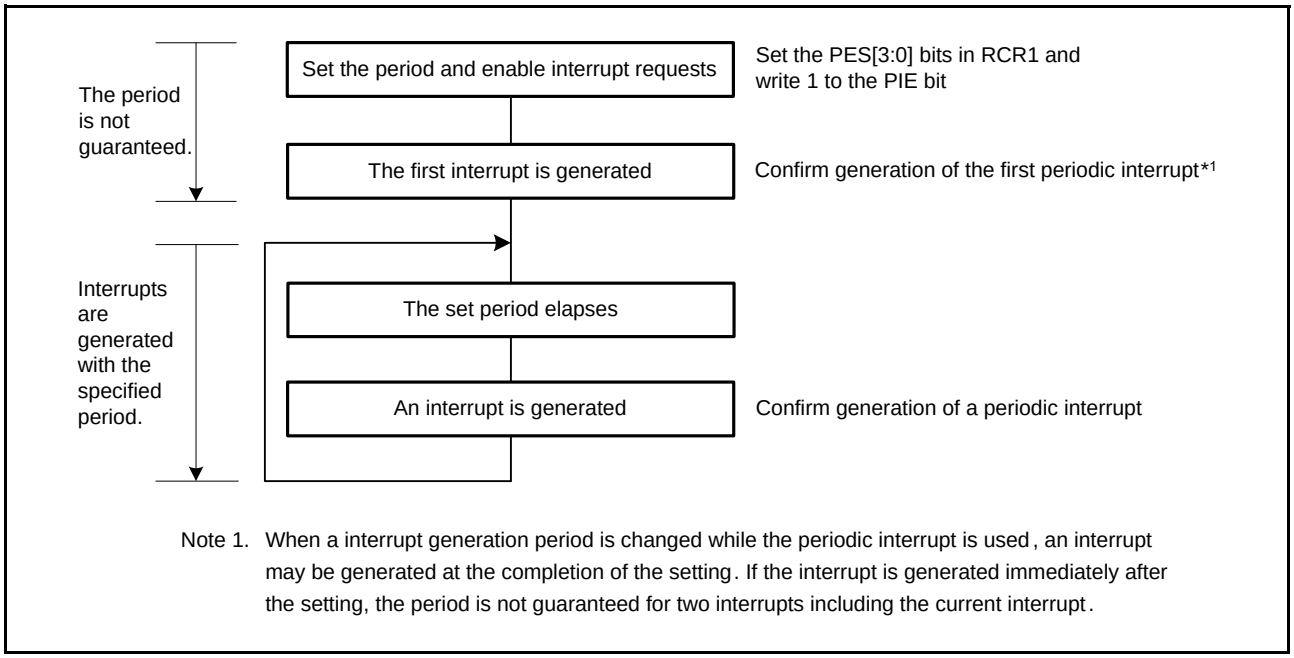


Figure 28.13 Using Periodic Interrupt Function

28.5.3 RTCOUT(1-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 Clock) output. When the time-error adjustment function is used, the period of RTCOUT(1-Hz Clock) output after adjustment is added or subtracted according to the adjustment value.

28.5.4 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 the battery back-up state) 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.

28.5.5 Points for Caution 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 28.3.5, Reading 64-Hz Counter and Time.
- When reading the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, registers RCR3 and RCR4, or frequency register after having written to the registers, perform three dummy read operations to ensure that the written value has been reflected in the register.
- When reading the RCR1.CIE and RCR2.RTCOE bits after having written to the register, the written value can be read.
- To read the correct value from the timer counter after return from a reset, or period in software standby mode, deep software standby mode, or the back-up state, wait for 1/128 second while the clock is operating (the RCR2.START bit = 1).

28.5.6 Procedure for Initializing the Realtime Clock without Using the Realtime Clock

Since the registers in the RTC are not initialized at reset time, depending on their initial state, unexpected interrupts may occur or counters may be activated unexpectedly, thus requiring larger amount of power.
Products that require no realtime clock should be initialized according to the initialization procedure shown in Figure 28.14.

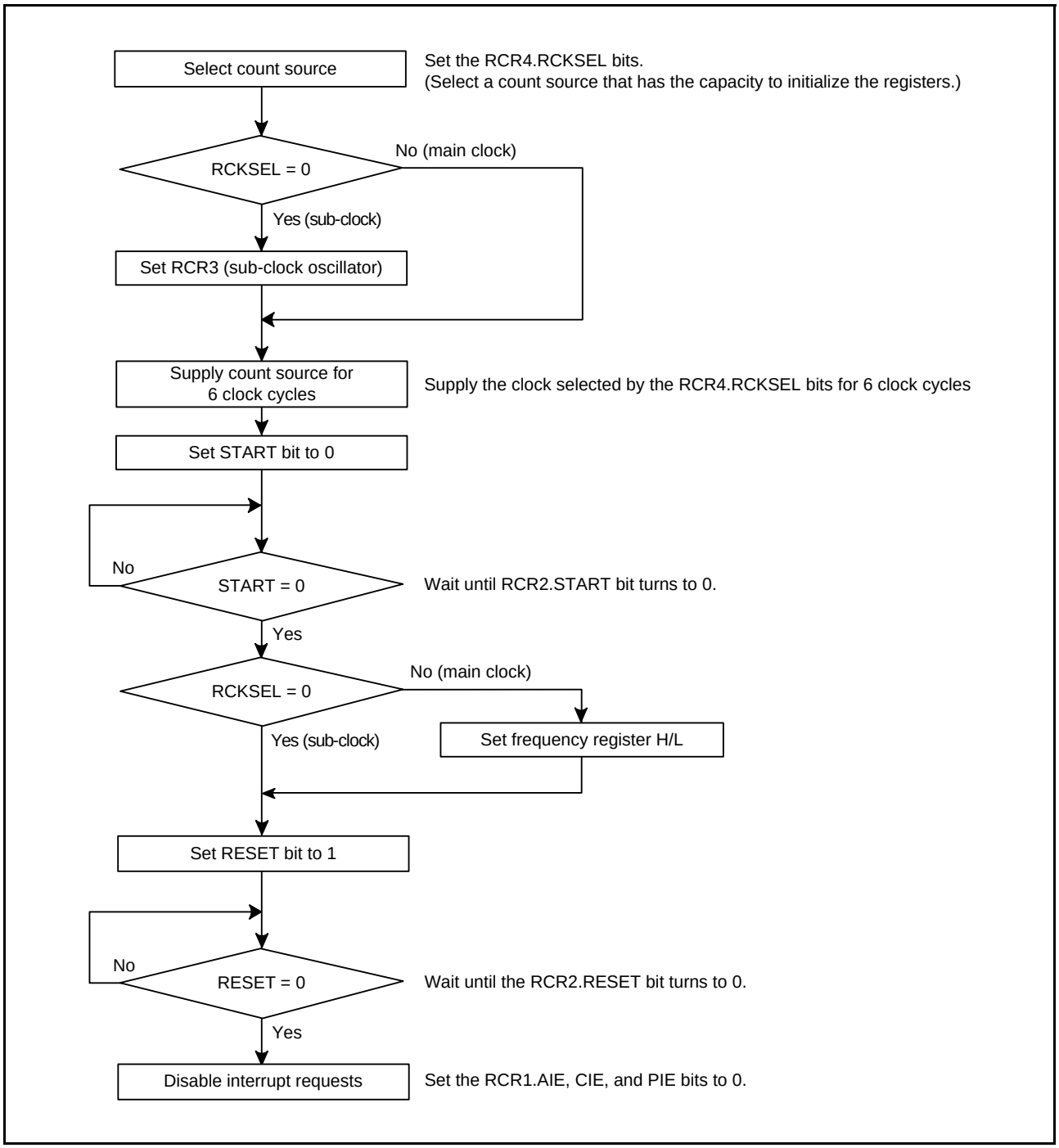


Figure 28.14 RTC Initialization Procedure

29. Watchdog Timer (WDTA)

The WDT has a 14-bit down-counter, and can be set up so that the chip is reset by a reset output when counting down from the initial value causes an underflow of the counter. Alternatively, generation of an interrupt request is selectable when the counter underflows. The initial value for counting can be restored to the down-counter by refreshing its value. The interval over which refreshing is possible can also be selected. Refreshing the counter during this interval will restore its initial value for counting, while attempting to refresh the counter beyond this interval leads to the output of a reset or interrupt request. The refresh interval can be adjusted and used to detect the program entering runaway conditions. The WDT stops counting after an underflow or an attempt at refreshing the counter beyond the allowed interval. Counting is restarted by refreshing the counter when the WDT is in register start mode. When the WDT is in auto-start mode, counting is restarted automatically after output of the reset or interrupt request.

29.1 Overview

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 the WDT (writing to the register).
In auto-start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, and reset or non-maskable interrupt request output at an underflow) should be made in the option function select register 0 (OFS0) before release from the reset state. In register start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, and reset or non-maskable interrupt request output at an underflow) should be made in the respective registers before the counter is started by refreshing after release from the reset state.
Set the WDT start mode select bit (OFS0.WDTSTRT bit) to select auto-start mode or register start mode.
When auto-start mode is selected (OFS0.WDTSTRT = 0), the WDT control register (WDTCR), and WDT reset control register (WDTRCR) settings are disabled and the settings of the OFS0 register are enabled.
When register start mode is selected (OFS0.WDTSTRT = 1), the settings of the OFS0 register are disabled and the WDTCR and WDTRCR settings are enabled.
Specifications of the WDT are listed in Table 29.1. Figure 29.1 is a block diagram of the WDT.

Table 29.1 Specifications of WDT (1/2)

Item	Specifications
Count source	Peripheral clock (PCLK)
Clock division ratio	Divide by 4, 64, 128, 512, 2,048, or 8,192
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 by refreshing the WDTRR register (writing 00h and then FFh) (register start mode)
Conditions for stopping the counter	• Pin reset (the down-counter and registers return to their initial values) • A counter underflows or a refresh error is generated Count restarts automatically in auto-start mode, or by refreshing the counter in register start mode.
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)
Interrupt request output sources	• A non-maskable interrupt (WUNI) is generated by an underflow of the down-counter • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the WDTSR register.
Output signal (internal signal)	• Reset output • Interrupt request output
Auto-start mode (controlled by the option function select register 0 (OFS0))	• Selecting the clock frequency division ratio after a reset (OFS0.WDTCKS[3:0] bits) • Selecting the time-out period of the watchdog timer (OFS0.WDTTOPS[1:0] bits) • Selecting the window start position in the watchdog timer (OFS0.WDTRPSS[1:0] bits) • Selecting the window end position in the watchdog timer (OFS0.WDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.WDTRSTIRQS bit)

Table 29.1 Specifications of WDT (2/2)

Item	Specifications
Register start mode (controlled by the WDT registers)	• Selecting the clock frequency division ratio after refreshing (WDTCR.CKS[3:0] bits) • Selecting the time-out period of the watchdog timer (WDTCR.TOPS[1:0] bits) • Selecting the window start position in the watchdog timer (WDTCR.RPSS[1:0] bits) • Selecting the window end position in the watchdog timer (WDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (WDTRCR.RSTIRQS bit)

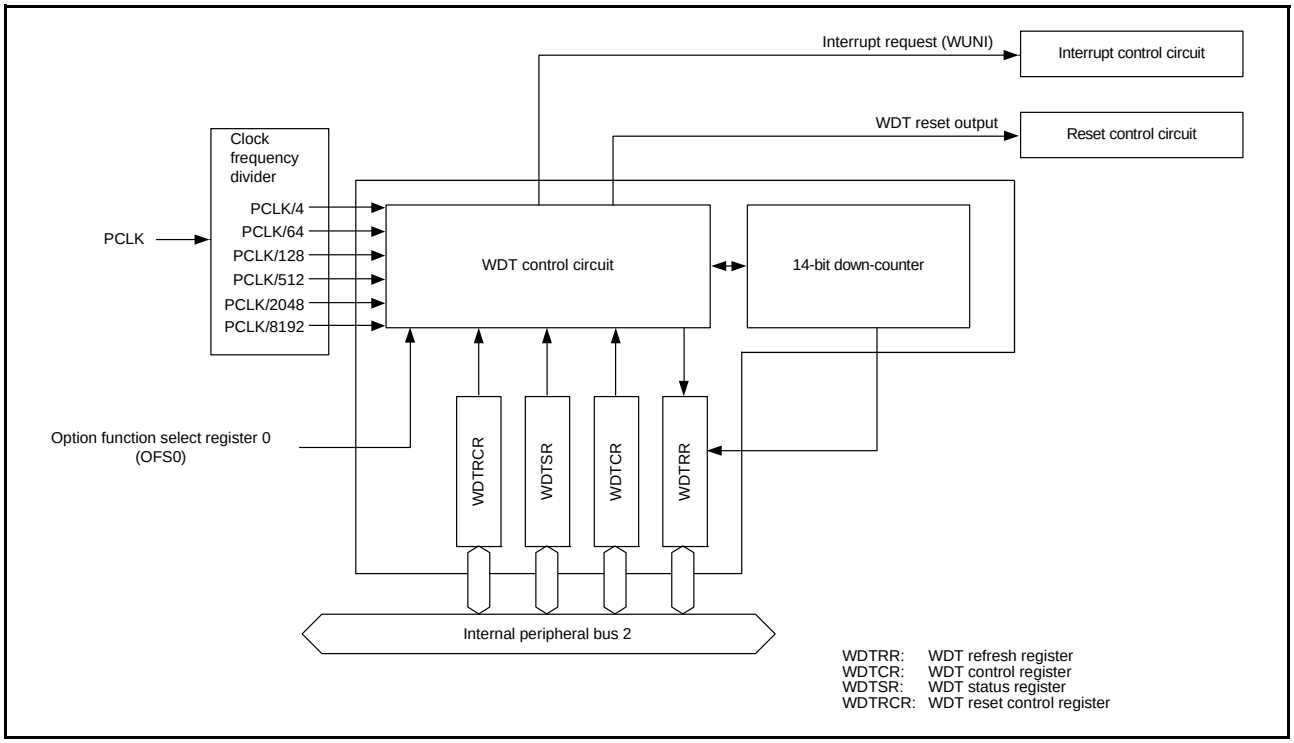
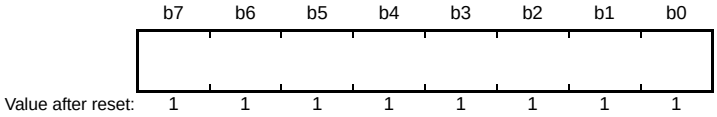


Figure 29.1 Block Diagram of WDT

29.2 Register Descriptions

29.2.1 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 time-out period selection (OFS0.WDTTOPS[1:0]) bits in option function select register 0 in auto-start mode. In register start mode, counting down starts from the value selected by setting the time-out period selection (WDTCR.TOPS[1:0]) bits in the WDT control register by 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 29.3.3, Refresh Operation.

29.2.2 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]	Time-Out Period Selection	b1 b0 0 0: 1,024 cycles (03FFh) 0 1: 4,096 cycles (0FFFh) 1 0: 8,192 cycles (1FFFh) 1 1: 16,384 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: PCLK/4 0 1 0 0: PCLK/64 1 1 1 1: PCLK/128 0 1 1 0: PCLK/512 0 1 1 1: PCLK/2048 1 0 0 0: PCLK/8192 Other settings 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 29.3.2, 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 the 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 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] Bits (Time-Out Period Selection)

The TOPS[1:0] bits select the time-out period (period until the down-counter underflows) from among 1,024, 4,096, 8,192, and 16,384 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 time-out period, and the number of PCLK cycles are listed in Table 29.2.

Table 29.2 Time-Out Period Settings

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Time-Out Period (Number of Cycles)	Cycles of PCLK Clock
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	PCLK/4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	PCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	PCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	PCLK/512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	PCLK/2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	PCLK/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 select the peripheral clock (PCLK) division ratio from among division by 4, 64, 128, 512, 2,048, and 8,192. 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 select 75%, 50%, 25%, or 0% of the count period for the window end position of the down-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 29.3 lists the counter values for the window start and end positions corresponding to the TOPS[1:0] bit values.

Table 29.3 Correspondence between Time-Out Period and Window Start and End Counter Values

TOPS[1:0] Bits		Time-Out 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 (Window Start Position Selection)

These bits select 100%, 75%, 50%, or 25% of the count period for the window start position of the down-counter (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 29.2 shows the relationship between the RPSS[1:0] and RPES[1:0] bit settings 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 ≥ window start setting, the window end setting is set to 0%.

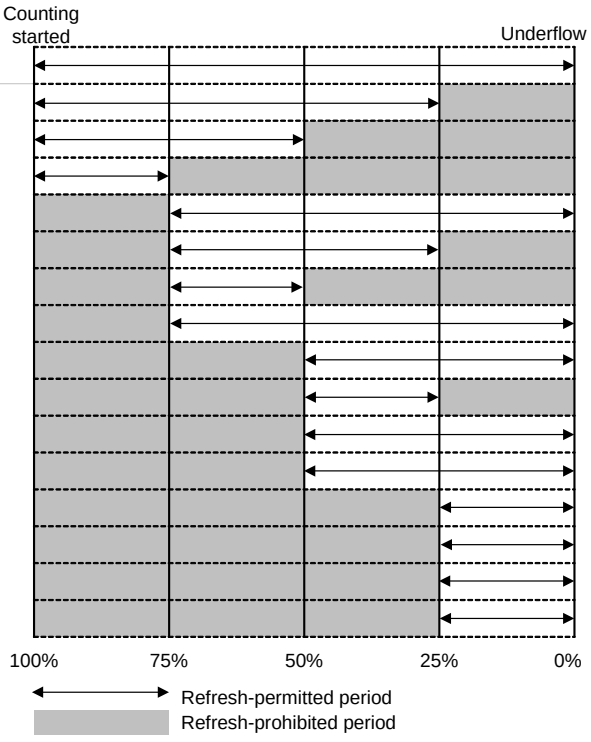


Figure 29.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

29.2.3 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.

WDTSR is initialized by the reset source of the WDT. WDTSR is not initialized by other reset sources.

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 bit 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 bit 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.
Clear the REFEF flag by writing 0 to it. Writing 1 has no effect.

29.2.4 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 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 29.3.2, Control over Writing to the WDTCR and WDTRCR Registers.
In auto-start mode, the WDTRCR register settings are disabled, and the settings in the option function select register 0 (OFS0) enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

29.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

29.3 Operation

29.3.1 Count Operation in Each Start Mode

Select the WDT start mode by setting the WDT start mode selection bit (OFS0.WDTSTRT) in the option function select register 0.
When the OFS0.WDTSTRT bit is 1 (register start mode), the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are enabled, and counting is started by refreshing (writing) the WDT refresh register (WDTRR).
When the OFS0.WDTSTRT bit is 0 (auto-start mode), the OFS0 register is enabled, and counting automatically starts after reset.

29.3.1.1 Register Start Mode

When the WDT start mode selection (OFS0.WDTSTRT) bit in the option function select register 0 is 1, register start mode is selected, and the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are enabled. After cancelling from the reset, set the clock division ratio, window start and end positions, and time-out 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 selected by setting the time-out period selection (WDTCR.TOPS[1:0]) bits.
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 WDT does not output the reset signal as long as this continues. However, if the down-counter underflows because the down-counter can not 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 (WUNI). Select reset output or interrupt request output by setting the reset interrupt request selection (WDTRCR.RSTIRQS) bit.

Figure 29.3 shows an example of operation under the following conditions.

- The WDT start mode selection (OFS0.WDTSTRT) bit is 1 (register start mode)
- The reset interrupt request selection (WDTRCR.RSTIRQS) bit is 1 (reset output is enabled)
- The window start position selection (WDTCR.RPSS[1:0]) bits are 10b (75%)
- The window end position selection (WDTCR.RPES[1:0]) bits are 10b (25%)

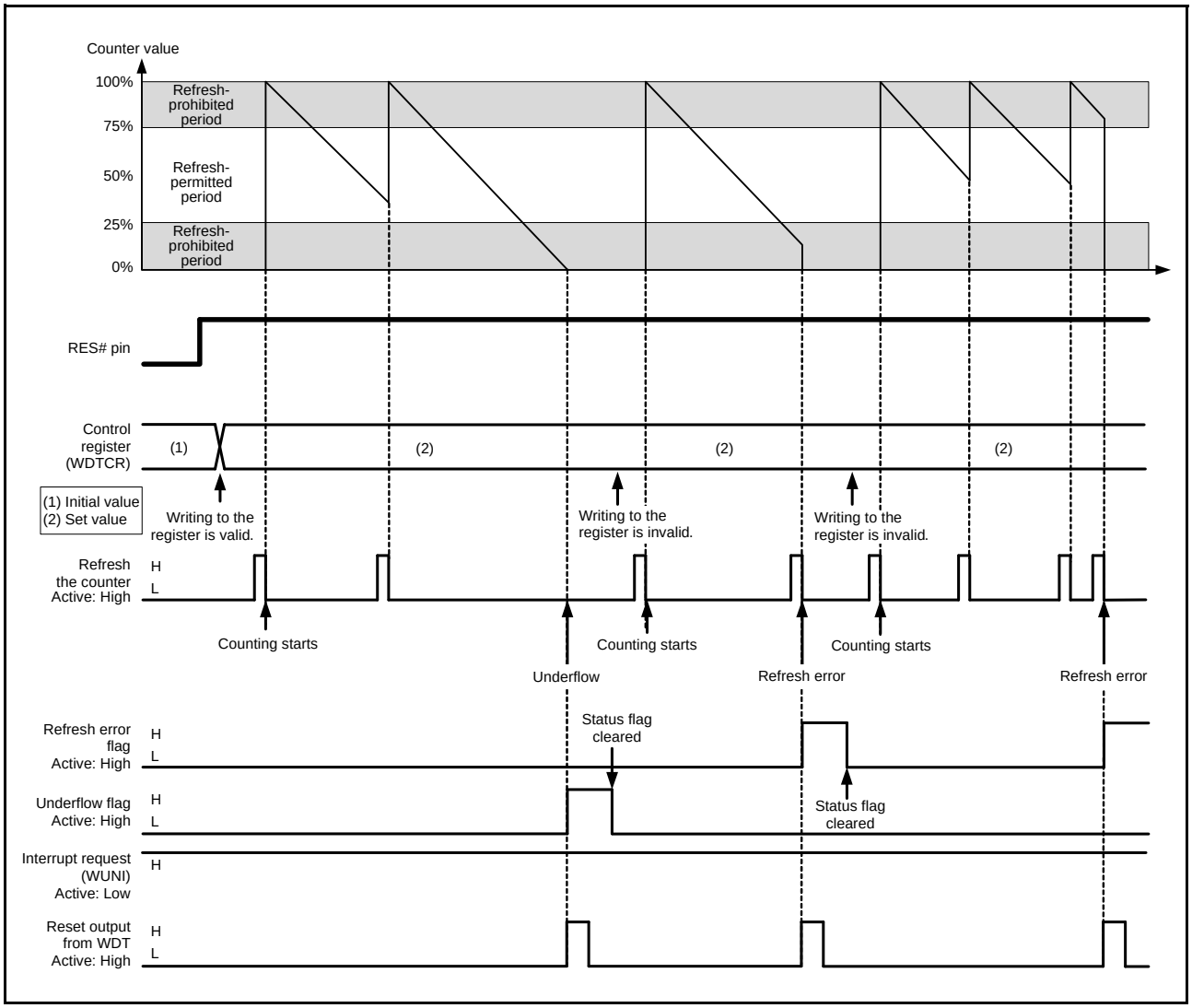


Figure 29.3 Operation Example in Register Start Mode

29.3.1.2 Auto-Start Mode

When the WDT start mode selection (OFS0.WDTSTRT) bit in the option function select register 0 is 0, auto-start mode is selected, and the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are disabled.
Within the reset state, the clock division ratio, window start and end positions, time-out period, and reset output or interrupt request output are set by the option function select register (OFS0). When the reset state is canceled, the down-counter automatically starts counting down from the value selected by the WDT time-out period selection (OFS0.WDTTOPS[1:0]) bits.
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 when 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 (WUNI). After output of the reset or non-maskable interrupt request for one cycle of counting, the down-counter is reloaded and counting is restarted when the time-out interval elapses. The reset output or interrupt request output can be selected through the WDT reset interrupt request selection (OFS0.WDTRSTIRQS) bit.

Figure 29.4 shows an example of operation under the following conditions.

- The WDT start mode selection (OFS0.WDTSTRT) bit is 0 (auto-start mode)
- The reset interrupt request selection (OFS0.WDTRSTIRQS) bit is 0 (non-maskable interrupt request output is enabled)
- The window start position selection (OFS0.WDTRPSS[1:0]) bits are 10b (75%)
- The window end position selection (OFS0.WDTRPES[1:0]) bits are 10b (25%)

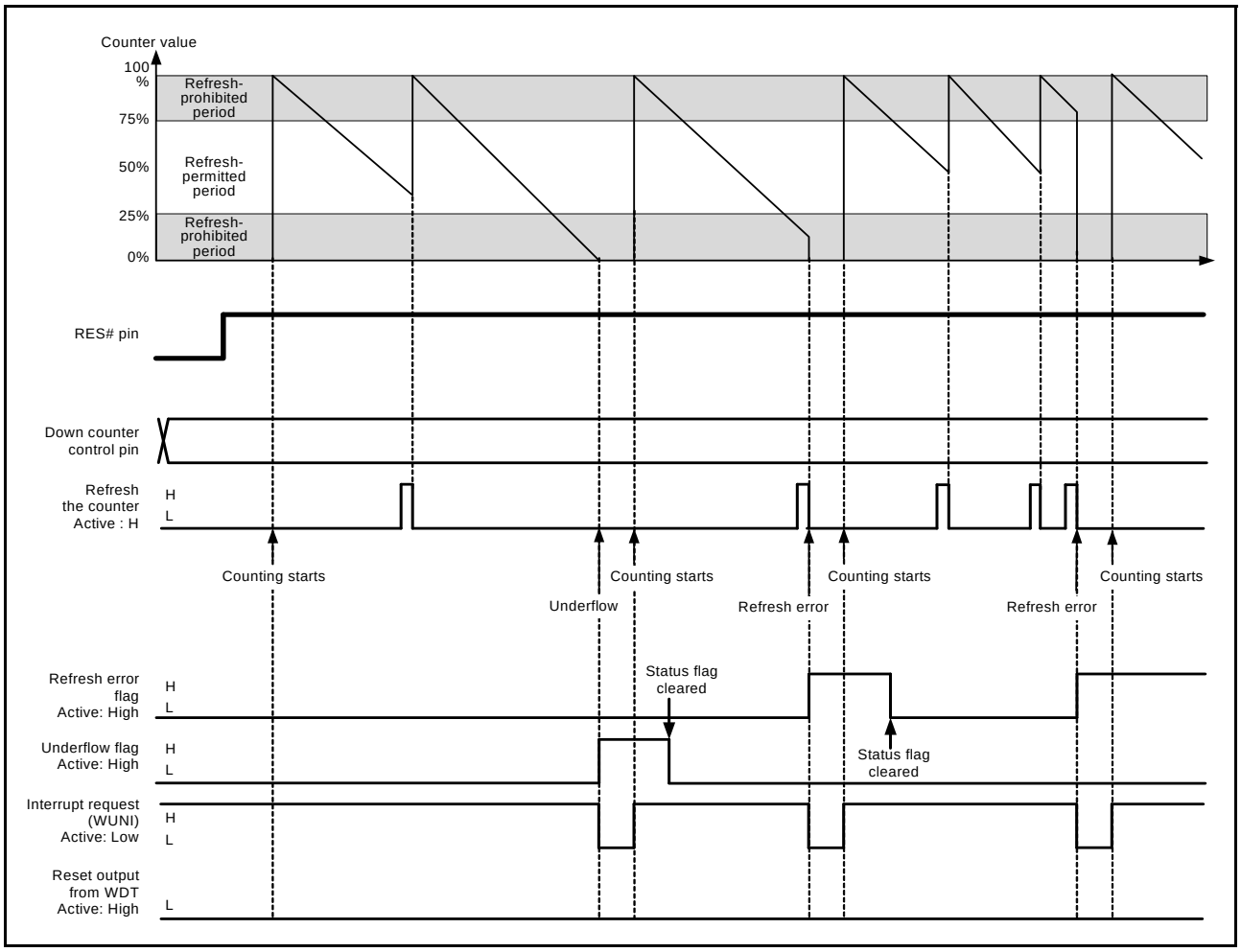


Figure 29.4 Operation Example in Auto-Start Mode

29.3.2 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 29.5 shows control waveforms produced in response to writing to the WDTCR.

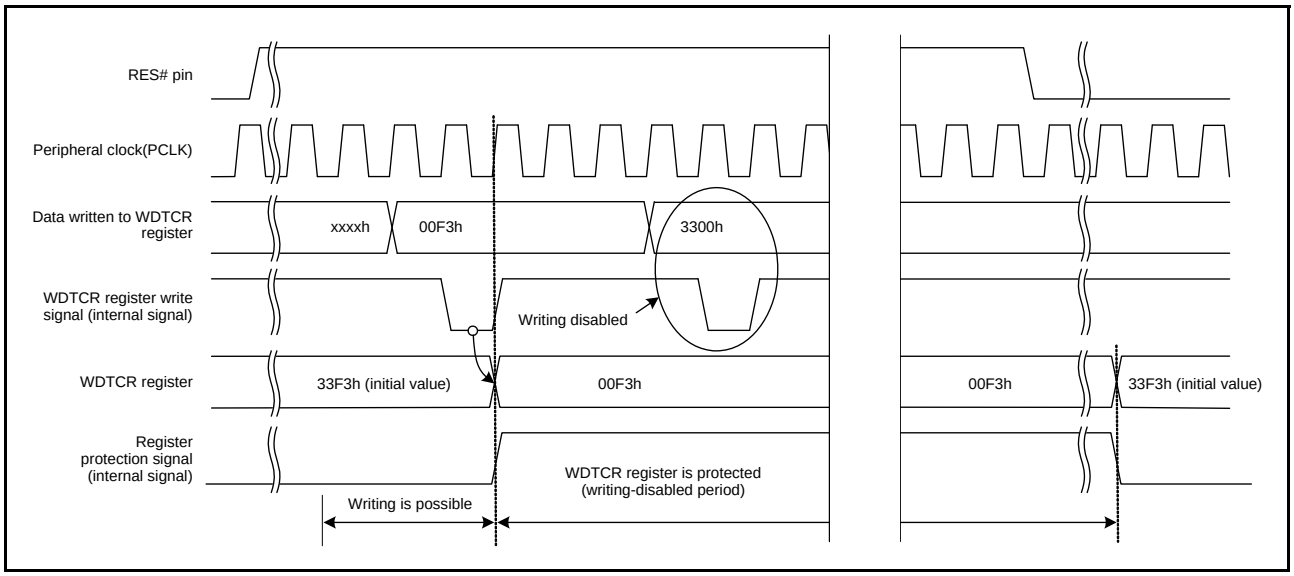


Figure 29.5 Control Waveforms Produced in Response to Writing to the WDTCR

29.3.3 Refresh Operation

The down-counter is refreshed and starts operation (counting is started by refreshing) 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 00h and then FFh to the WDT refresh register (WDTRR).
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 (n–th 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 WDTRR is accessed or WDTRR is read between writing 00h and writing FFh to WDTRR, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h→FFh
- 00h (n–1-th time)→00h (n-th 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

Even when 00h is written to WDTRR outside the refresh-permitted period, if FFh is written to WDTRR in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the WDTRR register, refreshing the down-counter requires up to four cycles of the signal for counting (the clock division ratio selection (WDTCR.CKS[3:0]) bits determine how many cycles of the peripheral clock (PCLK) make up one cycle for counting). Therefore, writing FFh to the WDTRR should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the down-counter can be checked by the counter bits (WDTSR.CNTVAL[13:0]).

[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to WDTRR before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to WDTRR after the value of the WDTSR.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 WDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to WDTRR.
- 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 WDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to WDTRR, no underflow occurs and refreshing is done.

Figure 29.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

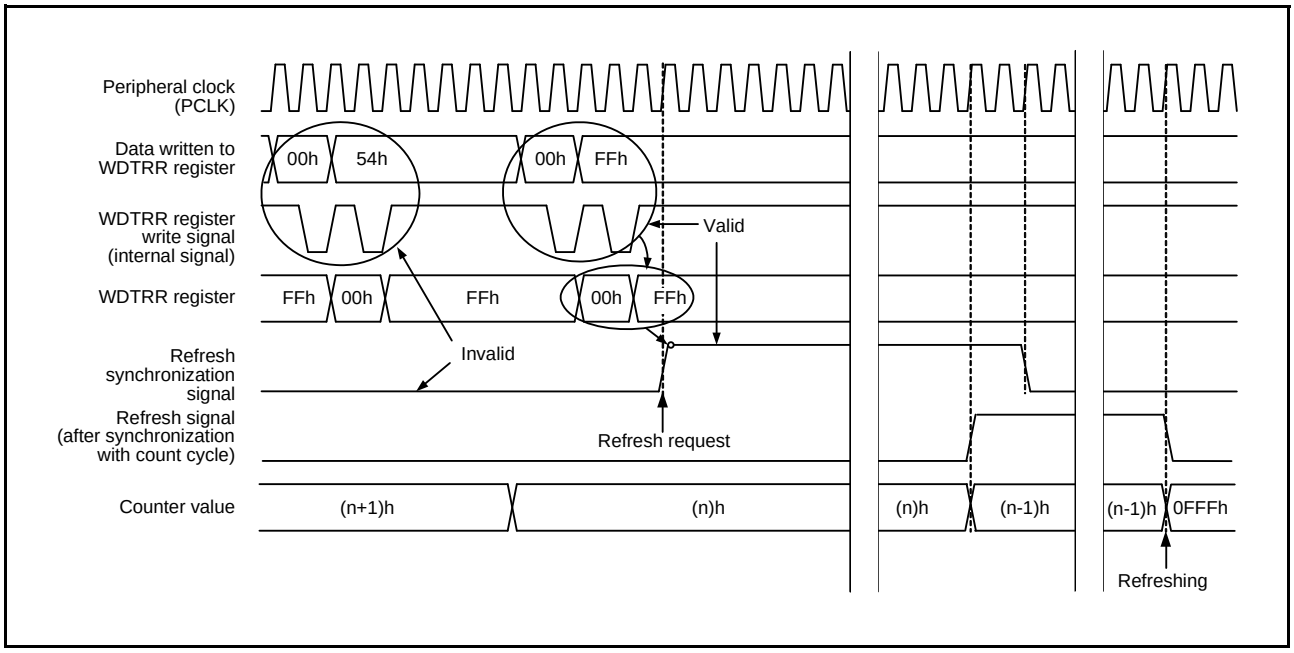


Figure 29.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

29.3.4 Status Flags

The refresh error (WDTSR.REFEF) and underflow (WDTSR.UNDFF) flags retain the source of the reset signal output from the WDT or the source of the interrupt request from the WDT.

Thus, after release from the reset state or interrupt request generation, read the WDTSR.REFEF and WDTSR.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 WDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.

In addition, several (at least five) cycles of the PCLK. are required to read the reflected value after clearing the flag by writing 0 to the bit.

29.3.5 Reset Output

When the reset interrupt selection (WDTRCR.RSTIRQS) bit is set to 1 in register start mode or when the WDT reset interrupt request selection (OFS0.WDTRSTIRQS) bit in the option function select register 0 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 cleared to 0) and kept in that state after assertion of the reset signal. After the reset is cancelled 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.

29.3.6 Interrupt Source

When the reset interrupt selection (WDTRCR.RSTIRQS) bit is set to 0 in register start mode or when the WDT reset interrupt request selection (OFS0.WDTRSTIRQS) bit in the option function select register 0 is set to 0 in auto-start mode, a non-maskable interrupt (WUNI) signal is output when an underflow in the down-counter or a refresh error occurs.

Table 29.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

29.3.7 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 29.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.

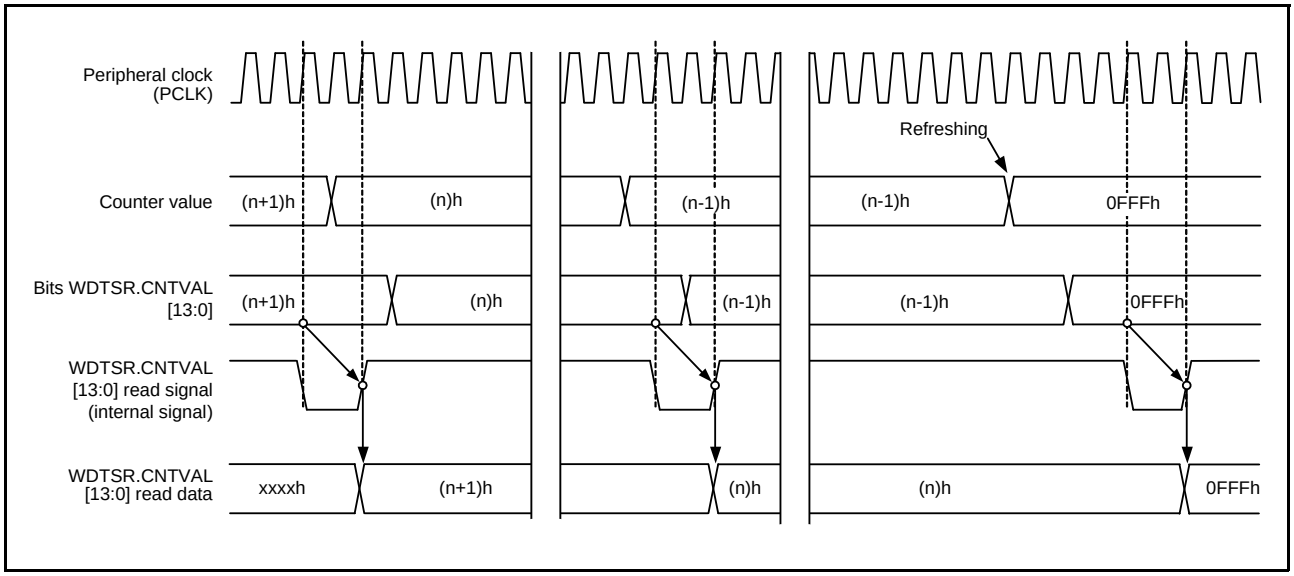


Figure 29.7 Processing for Reading WDT Down-Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

29.3.8 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 29.5 lists the correspondence between the option function select register 0 (OFS0) and the WDT registers (WDT control register (WDTCR) and WDT reset control register (WDTRCR)) regarding control of the down-counter and reset or interrupt request output. Control can be switched between the OFS0 register and the WDT registers (WDTCR and WDTRCR) through the setting of the WDT start mode selection (OFS0.WDTSTRT) bit.
Note that the OFS0 register setting should be kept unchanged during WDT operation.
For details on the OFS0 register, see section 7.2.1, Option Function Select Register 0 (OFS0).

Table 29.5 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Target of Control	Function	OFS0 Register (Effective in Auto-Start Mode) OFS0.WDTSTRT = 0	WDT Registers (Effective in Register Start Mode) OFS0.WDTSTRT = 1
Down-counter	Time-out 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

30. Independent Watchdog Timer (IWDTa)

The IWDT has a 14-bit down-counter, and can be set up so that the chip is reset by a reset output when counting down from the initial value causes an underflow of the counter. Alternatively, generation of an interrupt request is selectable when the counter underflows. The initial value for counting can be restored to the down-counter by refreshing its value. The interval over which refreshing is possible can also be selected. Refreshing the counter during this interval will restore its initial value for counting, while attempting to refresh the counter beyond this interval leads to the output of a reset or interrupt request. The refresh interval can be adjusted and used to detect the program entering runaway conditions. The IWDT stops counting after an underflow or an attempt at refreshing the counter beyond the allowed interval. Counting is restarted by refreshing the counter when the IWDT is in register start mode. When the IWDT is in auto-start mode, counting is restarted automatically after output of the reset or interrupt request.

30.1 Overview

The IWDT 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 the IWDT (writing to the register).
In auto-start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, reset or non-maskable interrupt request output at an underflow, and count stop control in sleep mode) should be made in the option function select register 0 (OFS0) before release from the reset state. In register start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, reset or non-maskable interrupt request output at an underflow, and count stop control in sleep mode) should be made in the respective registers before the counter is started by refreshing after release from the reset state.
Set the IWDT start mode select bit (OFS0.IWDTSTRT) to select auto-start mode or register start mode.
When auto-start mode is selected (OFS0.IWDTSTRT = 0), the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) settings are disabled and the OFS0 register settings are enabled.
When register start mode is selected (OFS0.IWDTSTRT = 1), the OFS0 register settings are ignored and the IWDTCR, IWDTRCR, and IWDTCSTPR settings take effect.
Specifications of the IWDT are listed in Table 30.1.

Table 30.1 Specifications of IWDT (1/2)

Item	Specifications
Count source	IWDT-dedicated low-speed clock (IWDTCLK)
Clock division ratio	Division 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 by refreshing the IWDTRR register (writing 00h and then FFh) (register start mode)
Conditions for stopping the counter	• Pin reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated Count restarts automatically in auto-start mode, or by refreshing the counter in register start mode
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)
Interrupt request output sources	• A non-maskable interrupt (WUNI) is generated by an underflow of the down-counter • When refreshing is done outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the IWDTSR register.
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep-mode count stop control output

Table 30.1 Specifications of IWDT (2/2)

Item	Specifications
Auto-start mode (controlled by the option function select register 0 (OFS0))	• Selecting the clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the time-out period of the watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the 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 division ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the time-out period of the watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the 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)

To use the IWDT, two clocks (peripheral clock (PCLK) and IWDT-dedicated low-speed clock (IWDTCLK)) should be supplied so that the IWDT works while the peripheral clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit down-counter and control circuits operate with IWDTCLK.
Signal lines between the blocks operating with the peripheral clock and IWDT-dedicated low-speed clock are connected through synchronization circuits.
Figure 30.1 is a block diagram of the IWDT

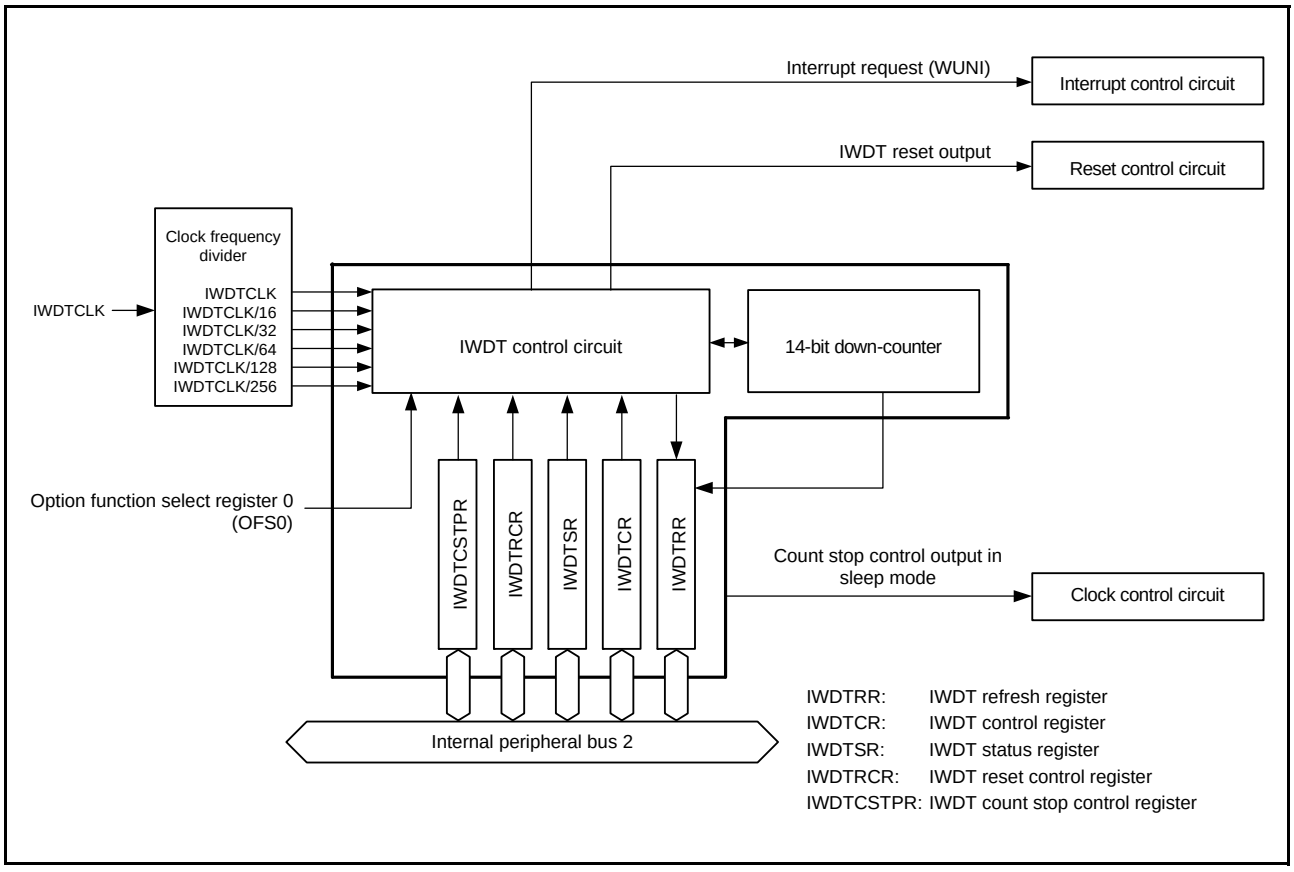
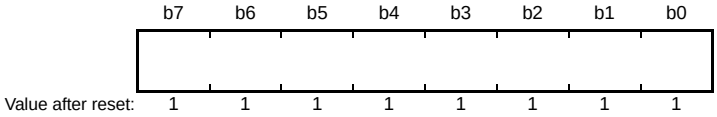


Figure 30.1 Block Diagram of IWDT

30.2 Register Descriptions

30.2.1 IWDT Refresh Register (IWDTRR)

Address(es): 0008 8030h



Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

IWDTRR refreshes the down-counter of the IWDT.
The down-counter of the IWDT is refreshed by writing 00h and then writing FFh to IWDTRR (refresh operation) within the refresh-permitted period.
After the down-counter has been refreshed, it starts counting down from the value selected by the IWDT time-out period selection (OFS0.IWDTTOPS[1:0]) bits in option function select register 0 in auto-start mode. In register start mode, counting down starts from the value selected by setting the time-out period selection (TOPS[1:0]) bits in the IWDT control register (IWDTCR) in the first refresh operation after release from the reset state.
When 00h is written, the read is 00h. When a value other than 00h is written, the read value FFh.
For details of the refresh operation, refer to section 30.3.3, Refresh Operation.

30.2.2 IWDT Control Register (IWDTCR)

Address(es): 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]	Time-Out Period Selection	b1 b0 0 0: 1,024 cycles (03FFh) 0 1: 4,096 cycles (0FFFh) 1 0: 8,192 cycles (1FFFh) 1 1: 16,384 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 0: IWDTCLK 0 0 1 0: IWDTCLK/16 0 0 1 1: IWDTCLK/32 0 1 0 0: IWDTCLK/64 1 1 1 1: IWDTCLK/128 0 1 0 1: IWDTCLK/256 Other settings 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 IWDTCR register. For details, refer to section 30.3.2, 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 the option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in the OFS0 register. For details, refer to section 30.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] Bits (Time-Out Period Selection)

The TOPS[1:0] bits select the time-out period (period until the down-counter underflows) from among 1,024, 4,096, 8,192, or 16,384 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 IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit settings, the time-out period, and the number of IWDTCLK cycles are listed in Table 30.2.

Table 30.2 Settings and Time-Out Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Division Ratio	Time-Out Period (Number of Cycles)	Cycles of IWDTCLK
0	0	0	0	0	0	IWDTCLK	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	0	1	0	0	0	IWDTCLK/16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	0	1	1	0	0	IWDTCLK/32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	0	0	0	0	IWDTCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	IWDTCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	0	1	0	0	IWDTCLK/256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

CKS[3:0] Bits (Clock Division Ratio Selection)

These bits select the IWDTCLK clock division ratio from among division by 1, 16, 32, 64, 128, and 256. Combined with the TOPS[1:0] bit setting, a count period between 1,024 and 4,194, 304 cycles of the IWDTCLK clock can be selected for the IWDT.

RPES[1:0] Bits (Window End Position Selection)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the down-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 30.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 30.3 Relationship between Time-Out Period and Window Start and End Counter Values

TOPS[1:0] Bits		Time-Out 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 Selection)

These bits select 100%, 75%, 50%, or 25% of the count period for the window start position for the down-counter (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 30.2 shows the relationship between 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 ≥ window start setting, the window end setting is set to 0%.

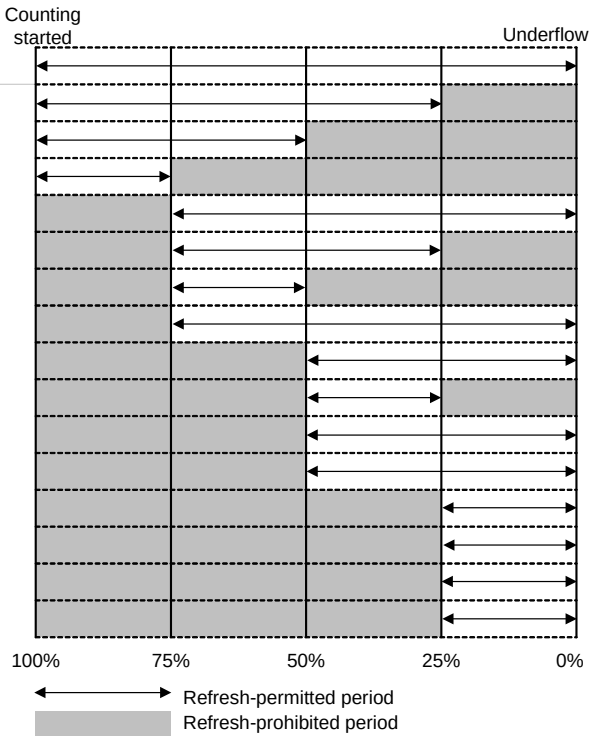


Figure 30.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

30.2.3 IWDT Status Register (IWDTSR)

Address(es): 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]	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.

IWDTSR is initialized by the reset source of the IWDT. IWDTSR is not initialized by other reset sources.

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 bit 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 bit 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.

30.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): 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 and cannot be modified.	R
b7	RSTIRQS	Reset Interrupt Request Selection	0: Non-maskable 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 30.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.
In auto-start mode, the IWDTRCR register settings are disabled, and the settings in the option function select register 0 (OFS0) enabled. The bit setting mode to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 30.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

30.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): 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 and cannot be modified.	R
b7	SLCSTP	Sleep-Mode Count Stop Control	0: Count stop function 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

There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 30.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.
In auto-start mode, the settings in the IWDTCSTPR register are ignored, and the settings in the option function select register 0 (OFS0) take effect. The bit setting mode to the IWDTCSTPR register can also be made in the OFS0 register. For details, refer to section 30.3.8, 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.

30.2.6 Option Function Select Register 0 (OFS0)

For the OFS0 register, refer to section 30.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

30.3 Operation

30.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDT start mode selection bit (OFS0.IWDTSTRT) in the 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 the OFS0 register is enabled, and counting automatically starts after reset.

30.3.1.1 Register Start Mode

When the IWDT start mode selection (OFS0.IWDTSTRT) bit in the 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 cancelling from the reset, set the clock division ratio, window start and end positions, and time-out period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the down-counter stop control at transitions to low-power-consumption modes in the IWDTCSTPR register. Then, refresh the down-counter to start counting down from the value selected by setting the time-out period selection (IWDTCR.TOPS[1:0]) bits. 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 down-counter underflows because the down-counter can not 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 (WUNI). Select reset output or interrupt request output by setting the reset interrupt request selection (IWDTRCR.RSTIRQS) bit.

Figure 30.3 shows an example of operation under the following conditions.

- The IWDT start mode selection (OFS0.IWDTSTRT) bit is 1 (register start mode)
- The reset interrupt request selection (IWDTRCR.RSTIRQS) bit is 1 (reset output is enabled)
- The window start position selection (IWDTCR.RPSS[1:0]) bits are 10b (75%)
- The window end position selection (IWDTCR.RPES[1:0]) bits are 10b (25%)

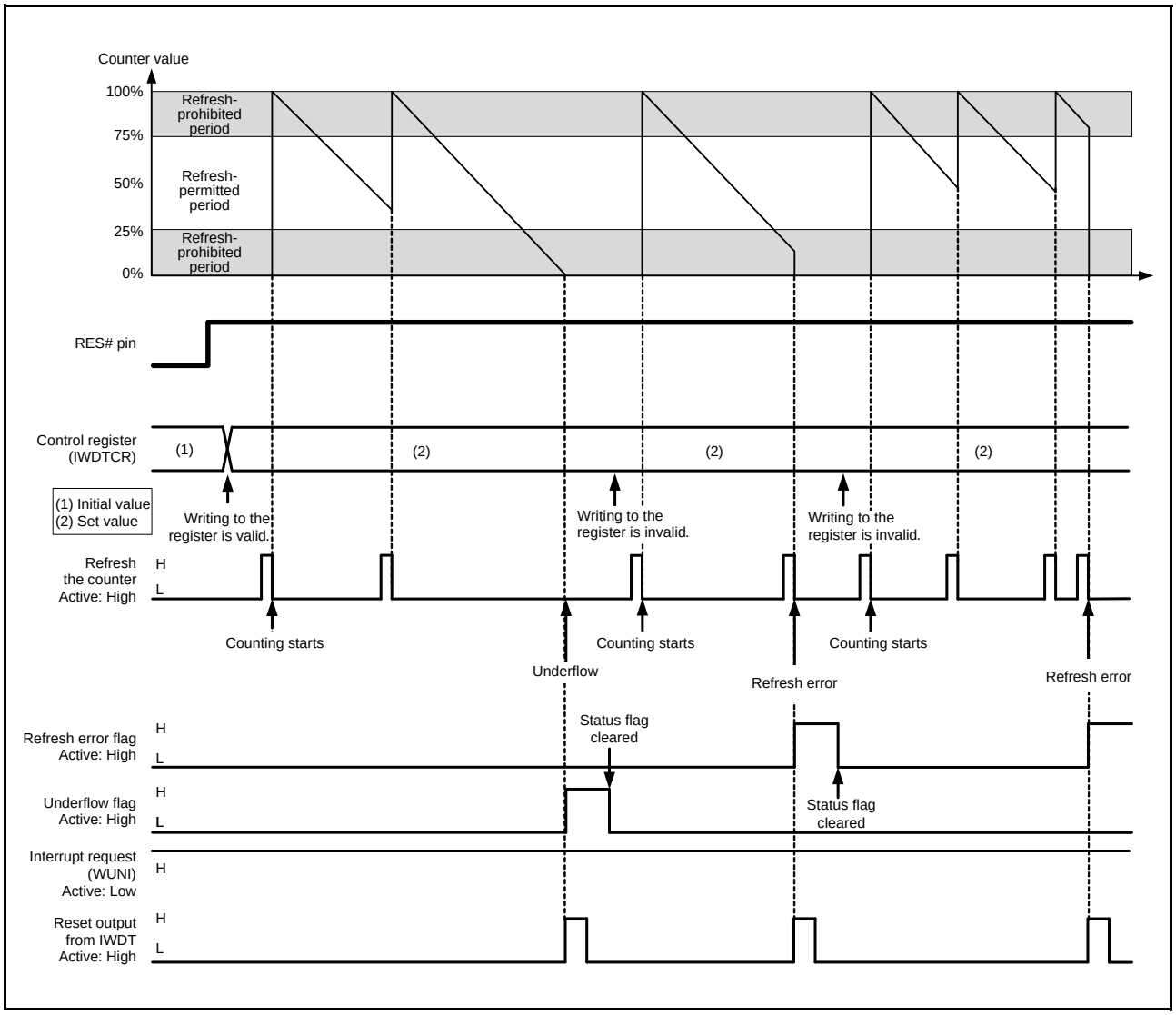


Figure 30.3 Operation Example in Register Start Mode

30.3.1.2 Auto-Start Mode

When the IWDT start mode selection (OFS0.IWDTSTRT) bit in the 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 division ratio, window start and end positions, time-out period, reset output or interrupt request output, and down-counter stop control at transitions to low-power-consumption modes should be specified in the OFS0 register. When the reset state is canceled, the down-counter automatically starts counting down from the value selected by the IWDT time-out period selection (OFS0.IWDTTOPS[1:0]) bits.

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 when the counter is refreshed and counting down continues. The IWDT 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 IWDT outputs the reset signal or non-maskable interrupt request (WUNI). After output of the reset or non-maskable interrupt request for one cycle of counting, the down-counter is reloaded and counting is restarted when the time-out interval elapses. The reset output or interrupt request output can be selected through the IWDT reset interrupt request selection (OFS0.IWDTRSTIRQS) bit.

Figure 30.4 shows an example of operation under the following conditions.

- The IWDT start mode selection (OFS0.IWDTSTRT) bit is 0 (auto-start mode)
- The reset interrupt request selection (OFS0.IWDTRSTIRQS) bit is 0 (non-maskable interrupt request output is enabled)
- The window start position selection (OFS0.IWDTRPSS[1:0]) bits are 10b (75%)
- The window end position selection (OFS0.IWDTRPES[1:0]) bits are 10b (25%)

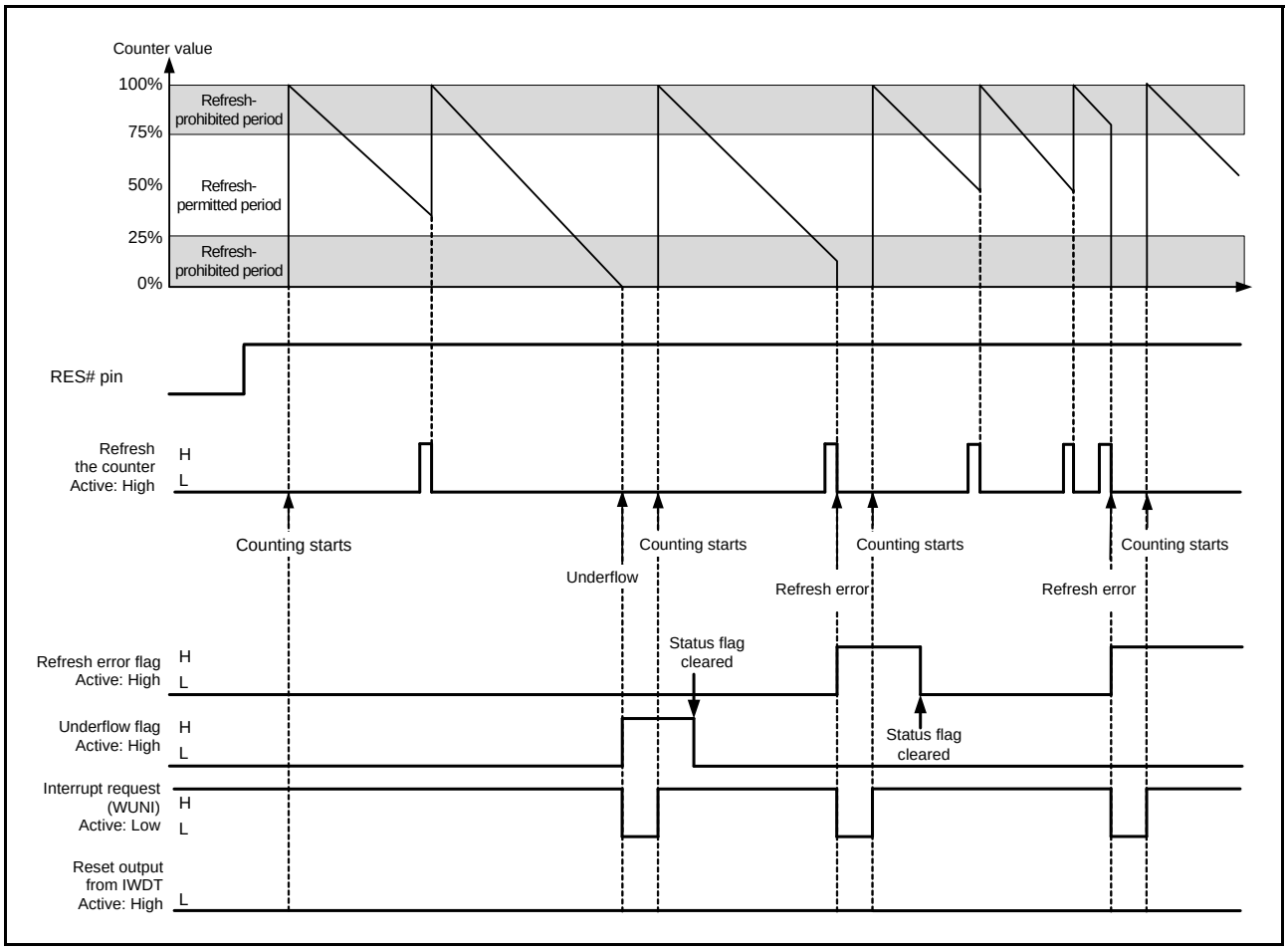


Figure 30.4 Operation Example in Auto-Start Mode

30.3.2 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 IWDTCR, IWDTRCR, or IWDTCSTPR is written to, the protection signal in the IWDT becomes 1 to protect 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 30.5 shows control waveforms produced in response to writing to the IWDTCR.

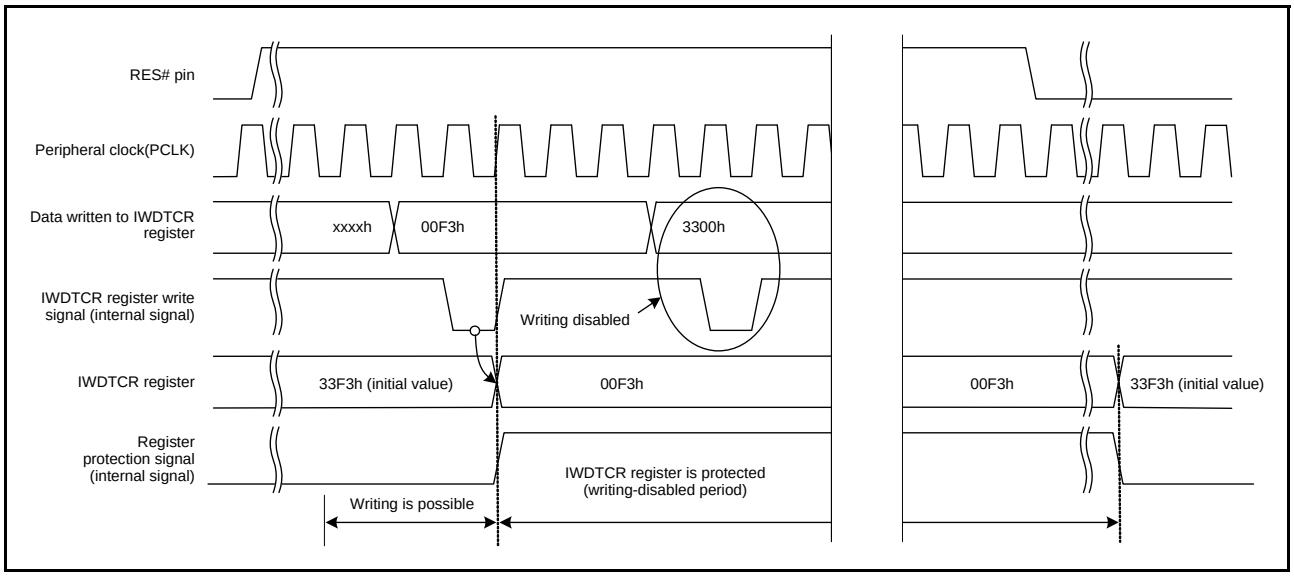


Figure 30.5 Control Waveforms Produced in Response to Writing to the IWDTCR

30.3.3 Refresh Operation

The down-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 down-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 (n–th 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 IWDTRR is accessed or IWDTRR is read between writing 00h and writing FFh to IWDTRR, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h→FFh
- 00h (n–1-th time) →00h (n-th time) →FFh
- 00h→access to another register or read from IWDTRR→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 IWDTRR outside the refresh-permitted period, if FFh is written to IWDTRR 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 down-counter requires up to four cycles of the signal for counting (the clock division ratio selection (the clock division ratio selection (IWDTCR.CKS[3:0]) bits determine how many cycles of the IWDT-dedicated low-speed clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the down-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 IWDTRR before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to IWDTRR 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 IWDTRR.
- 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 IWDTRR, no underflow occurs and refreshing is done.

Figure 30.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock division ratio = IWDTCLK, and Figure 30.7 shows those when PCLK < IWDTCLK and clock division ratio = IWDTCLK/16.

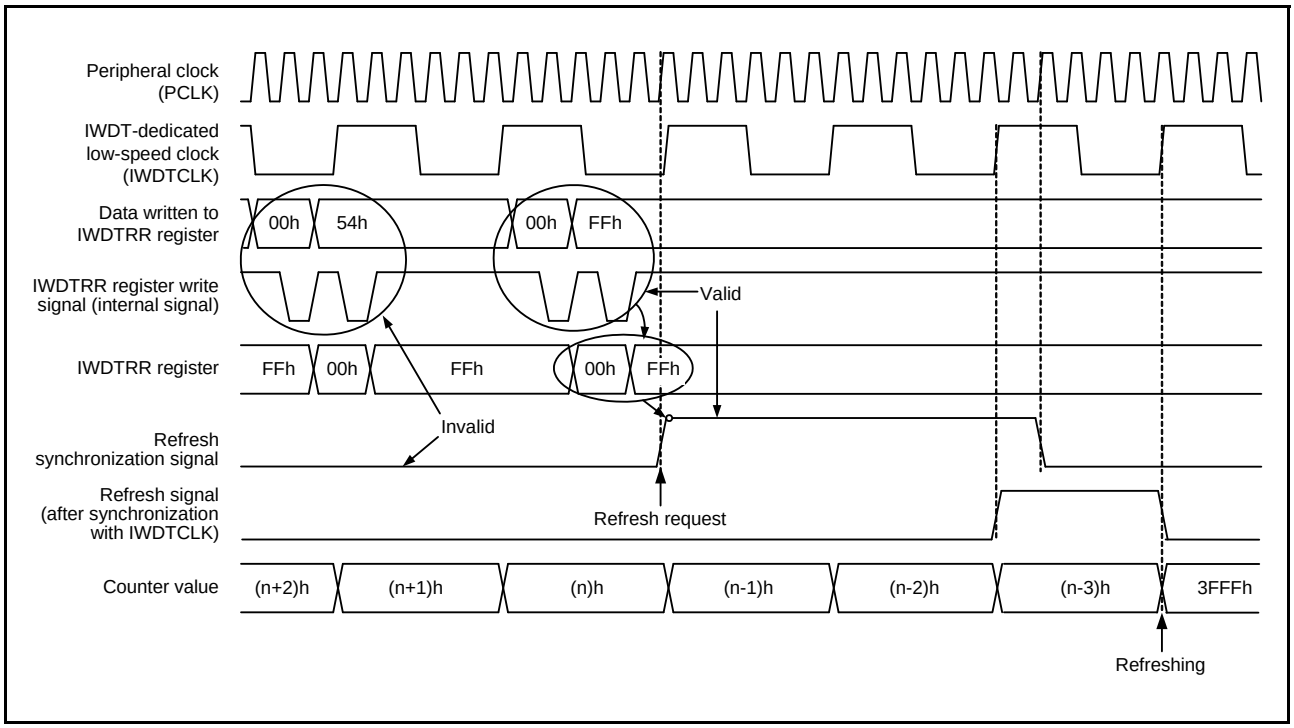


Figure 30.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

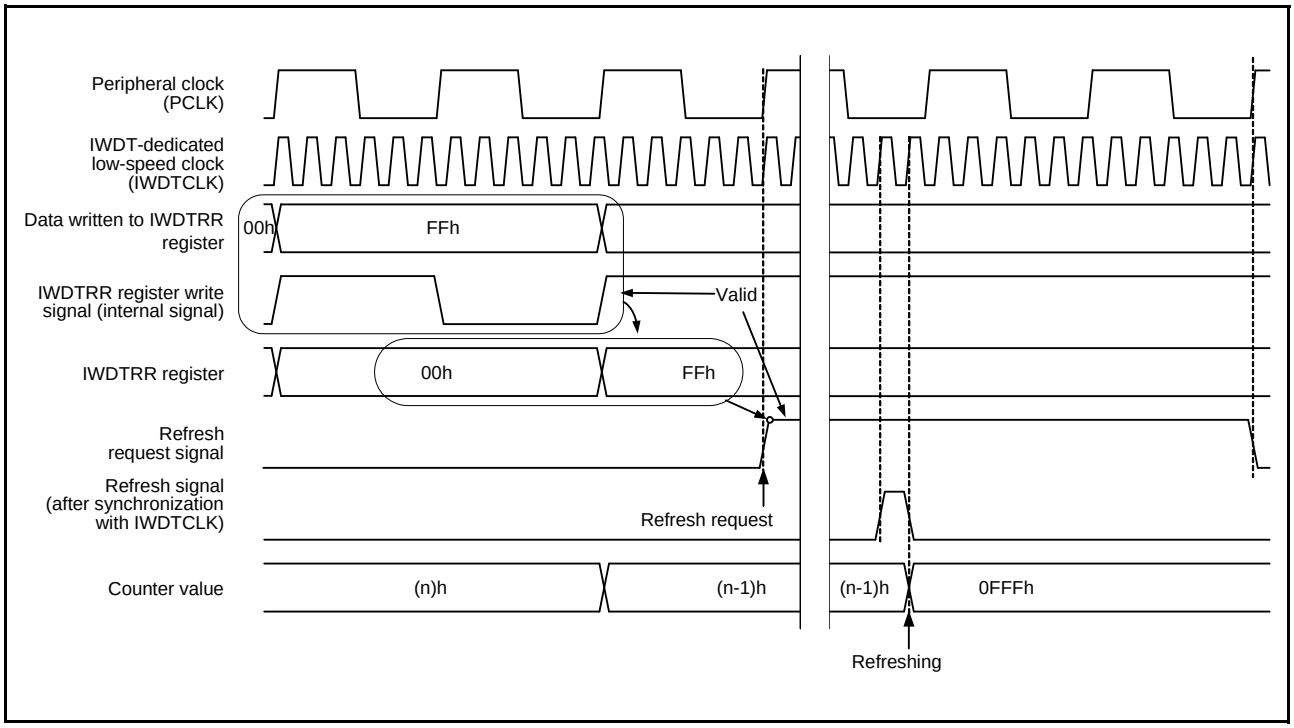


Figure 30.7 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 01b)

30.3.4 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.

In addition, several cycles of the IWDTCLK (at least three) and of the PCLK (at least two) are required to read the reflected value after clearing the flag by writing 0 to the bit.

30.3.5 Reset Output

When the reset interrupt selection (IWDTRCR.RSTIRQS) bit is set to 1 in register start mode or when the IWDT reset interrupt request selection (OFS0.IWDTRSTIRQS) bit in the option function select register 0 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 cleared to 0) and kept in that state after assertion of the reset signal. After the reset is cancelled 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.

30.3.6 Interrupt Source

When the reset interrupt selection (IWDTRCR.RSTIRQS) bit is set to 0 in register start mode or when the IWDT reset interrupt request selection (OFS0.IWDTRSTIRQS) bit in the option function select register 0 is set to 0 in auto-start mode, a non-maskable interrupt (WUNI) signal is output when an underflow in the down-counter or a refresh error occurs.

Table 30.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

30.3.7 Reading the Down-Counter Value

As the down-counter in IWDT-dedicated low-speed clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLK) and stores it in the down-counter value (IWDTSR.CNTVAL[13:0]) bits of the IWDT status register.
Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.
Reading the down-counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual down-counter value by a value of one count.
Figure 30.8 shows the processing for reading the IWDT down-counter value when PCLK > IWDTCLK and clock division ratio = IWDTCLK, and Figure 30.9 shows the processing when PCLK < IWDTCLK and clock division ratio = IWDTCLK/16.

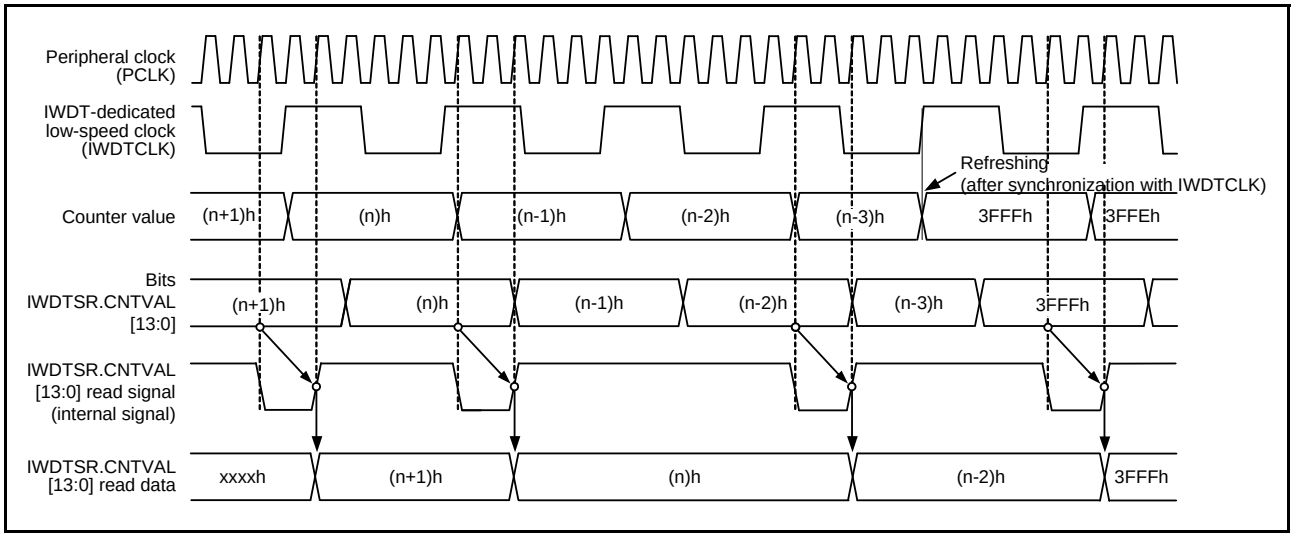


Figure 30.8 Processing for Reading IWDT Down-Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

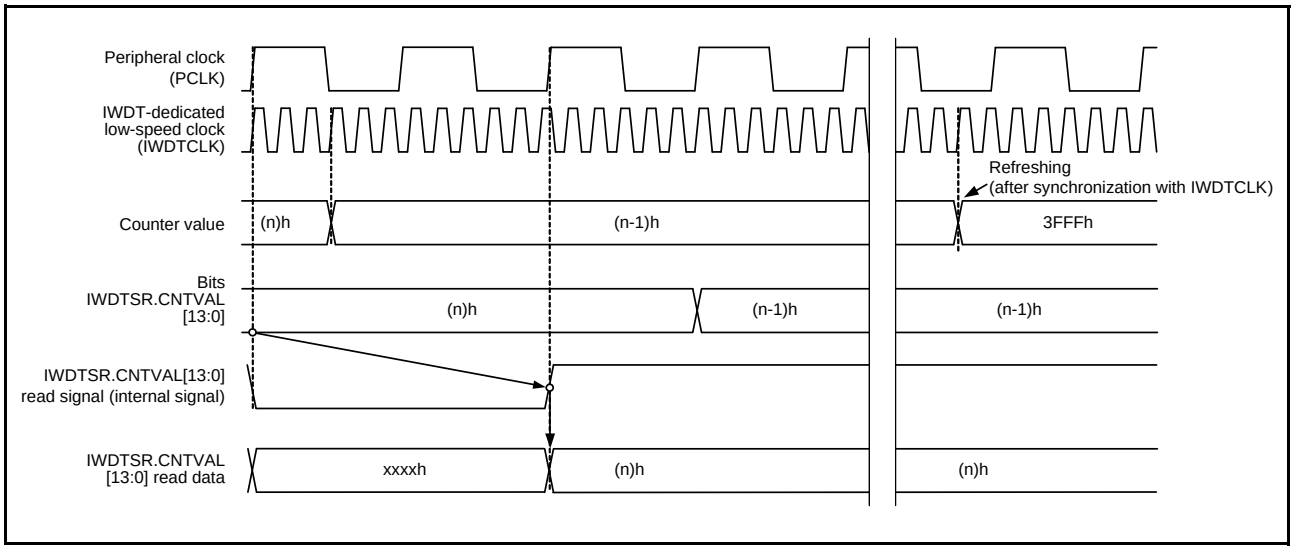


Figure 30.9 Processing for Reading IWDT Down-Counter Value (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 11b)

30.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 30.5 lists the correspondence between the option function select register 0 (OFS0) and the IWDT registers (IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR)) regarding control of the down-counter, reset or interrupt request output, and count stop function. Control can be switched between the option function select register 0 (OFS0) and the IWDT registers (IWDTCR, IWDTRCR, and IWDTCSTPR) through the setting of the IWDT start mode selection (OFS0.IWDTSTRT) bit in the option function select register 0 (OFS0).
Note that the option function select register 0 (OFS0) setting should be kept unchanged during IWDT operation.
For details on the option function select register 0 (OFS0), see section 7.2.1, Option Function Select Register 0 (OFS0).

Table 30.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Effective in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Effective in Register Start Mode) OFS0.IWDTSTRT = 1
Down-counter	Time-out period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock division 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 selection	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

30.4 Usage Notes

30.4.1 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.

31. Ethernet Controller (ETHERC)

31.1 Overview

The RX63N Group has an on-chip Ethernet controller (ETHERC) conforming to the Ethernet or IEEE802.3 MAC (Media Access Control) layer standard. Connecting a physical-layer LSI (PHY-LSI) complying with this standard enables the ETHERC to perform transmission and reception of Ethernet/IEEE802.3 frames. The ETHERC has one MAC layer interface port. The ETHERC is connected to the Ethernet direct memory access controller for Ethernet controller (EDMAC) inside this LSI, and carries out high-speed data transfer to and from the memory.
Table 31.1 shows the specifications of the ETHERC, Figure 31.1 shows a configuration of the ETHERC, and Table 31.2 shows the pin configuration of the ETHERC.

Table 31.1 Specifications of ETHERC

Item	Description
Protocol	• Flow control conforming to IEEE802.3x
Data transmission and reception	• Transmission and reception of Ethernet/IEEE802.3 frames
Transfer rate	• Supports 10- and 100-Mbps transfer
Mode	• Supports full-duplex and half-duplex modes
Interface	• Conforms to the MII (Media Independent Interface) and RMII (Reduced Media Independent Interface) specifications of the IEEE802.3u standard
Function	• Magic Packet*1 detection and Wake-On-LAN (WOL) signal output

Note 1. Magic Packet™ is a registered trademark of Advanced Micro Devices, Inc.

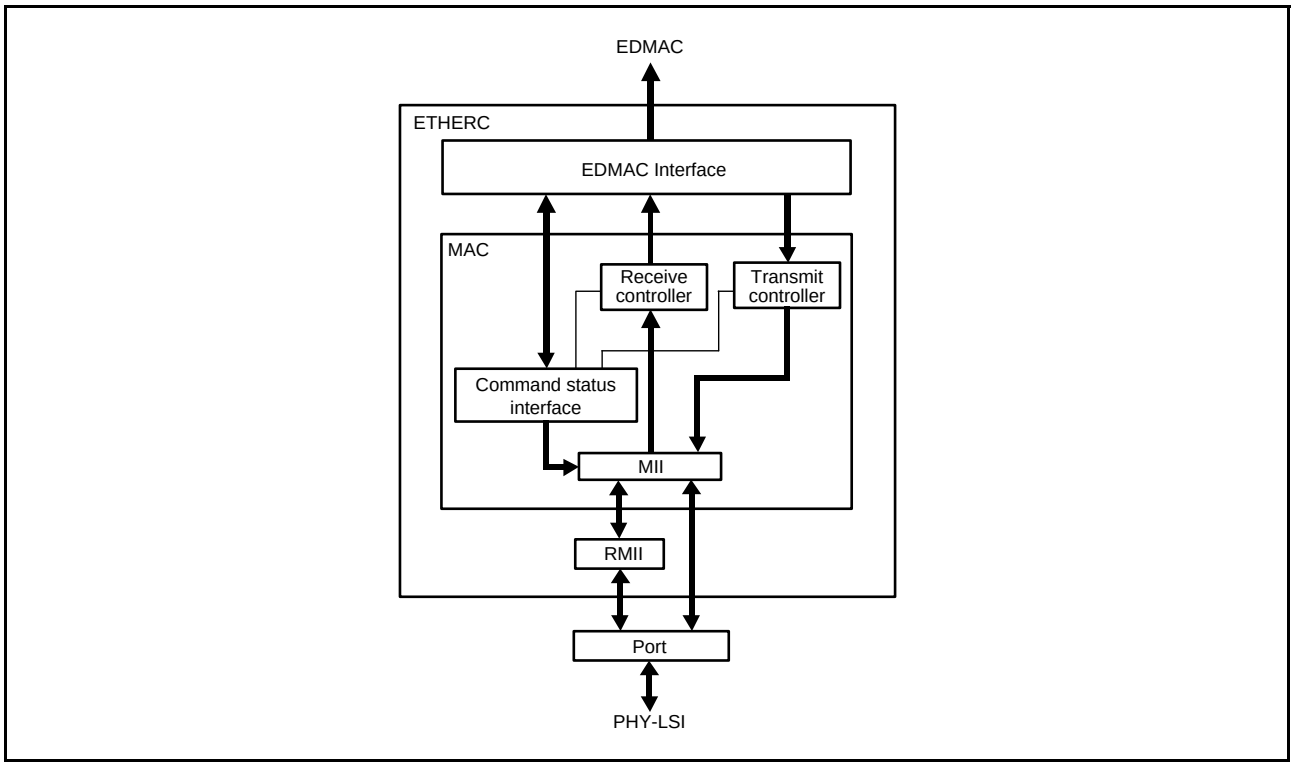


Figure 31.1 Configuration of ETHERC

Table 31.2 Pin Configuration of ETHERC

Operating Mode	Name	I/O	Function
MII	ET_TX_CLK*1	Input	Transmit clock ET_TX_EN, ET_ETXD3 to ET_ETXD0, and ET_TX_ER timing reference signal
	ET_RX_CLK*1	Input	Receive clock ET_RX_DV, ET_ERXD3 to ET_ERXD0, and ET_RX_ER timing reference signal
	ET_TX_EN*1	Output	Transmit enable signal Indicates that transmit data is ready on ET_ETXD3 to ET_ETXD0
	ET_ETXD3 to ET_ETXD0*1	Output	4-bit transmit data
	ET_TX_ER*1	Output	Sends error state occurred during data reception to the PHY-LSI
	ET_RX_DV*1	Input	Indicates that valid receive data is on ET_ERXD3 to ET_ERXD0
	ET_ERXD3 to ET_ERXD0*1	Input	4-bit receive data
	ET_RX_ER*1	Input	Receive error Identifies error state occurred during data reception
	ET_CRS*1	Input	Carrier detection signal
	ET_COL*1	Input	Collision detection signal
	ET_MDC*1	Output	Reference clock signal for information transfer via ET_MDIO
	ET_MDIO*1	I/O	Bidirectional signal for exchange of management information between this STA and PHY-LSI
	ET_LINKSTA	Input	Inputs link status from PHY-LSI
	ET_EXOUT	Output	External output pin
	ET_WOL	Output	Wake-On-LAN signal indicating reception of Magic Packet™
RMII	ET_MDC	Output	Reference clock signal for information transfer via ET_MDIO
	ET_MDIO	I/O	Bidirectional signal for exchange of management information between this STA and PHY-LSI
	ET_WOL	Output	Wake-On-LAN signal indicating reception of Magic Packet™
	ET_LINKSTA	Input	Inputs link status from PHY-LSI
	ET_EXOUT	Output	External output pin
	REF50CK*2	Input	RMII_TXD_EN, RMII_TXD1 to RMII_TXD0, RMII_CRS_DV, RMII_RXD1 to RMII_RXD0, and RMII_RX_ER timing reference signal
	RMII_TXD1 to RMII_TXD0*2	Output	2-bit transmit data
	RMII_TXD_EN*2	Output	Indicates that transmit data is ready on RMII_TXD1 and RMII_TXD0
	RMII_RXD1 to RMII_RXD0*2	Input	2-bit receive data
	RMII_RX_ER*2	Input	Identifies error state occurred during data reception
	RMII_CRS_DV*2	Input	Carrier detection signal/indication of valid receive data on RMII_RXD1 and RMII_RXD0

Note 1. MII signal conforming to IEEE802.3u
Note 2. RMII signal conforming to IEEE802.3u

31.2 Register Descriptions

31.2.1 ETHERC Mode Register (ECMR)

Address(es): 000C 0100h

	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: ETHERC performs normal operation 1: ETHERC performs promiscuous mode operation	R/W
b1	DM	Duplex Mode	0: Half-duplex transfer is specified. 1: Full-duplex transfer is specified.	R/W
b2	RTM	Transmission/Reception Rate	0: 10 Mbps 1: 100 Mbps	R/W
b3	ILB	Internal Loop Back Mode	0: Normal data transmission/reception is performed. 1: Data loopback is performed inside the MAC in the ETHERC when DM = 1.	R/W
b4	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b5	TE	Transmission Enable	0: Transmitting function is disabled. 1: Transmitting function is enabled.	R/W
b6	RE	Reception Enable	0: Receiving function is disabled. 1: Receiving function is enabled.	R/W
b8, b7	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b9	MPDE	Magic Packet™ Detection Enable	0: Magic Packet™ detection is not enabled. 1: Magic Packet™ detection is enabled.	R/W
b11, b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b12	PRCEF	CRC Error Frame Reception Enable	0: A frame with a CRC error is received as a frame with an error. 1: A frame with a CRC error is received as a frame without an error.	R/W
b15 to b13	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b16	TXF	Operating Mode for Transmitting Port Flow Control	0: PAUSE frame detection is disabled. (Automatic PAUSE frame is not transmitted) 1: Flow control for the transmitting port is enabled. (Automatic PAUSE frame is transmitted as required)	R/W
b17	RXF	Operating Mode for Receiving Port Flow Control	0: PAUSE frame detection is disabled 1: Flow control for the receiving port is enabled	R/W
b18	PFR	PAUSE Frame Receive Mode	0: PAUSE frame is not transferred to EDMAC 1: PAUSE frame is transferred to EDMAC	R/W
b19	ZPF	PAUSE Frame Usage with TIME = 0 Enable	0: Control of a PAUSE frame whose TIME parameter value is 0 is disabled 1: Control of a PAUSE frame whose TIME parameter value is 0 is enabled	R/W
b20	TPC	PAUSE Frame Transmission	0: PAUSE frame is not transmitted in a PAUSE period 1: PAUSE frame is transmitted even in a PAUSE period	R/W
b31 to b21	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ECMR specifies the operating mode of the ETHERC.
The settings in ECMR should be made in the initialization process after a reset.
The operating mode setting must not be changed while the transmitting and receiving functions are enabled. To switch the operating mode, return the ETHERC and EDMAC to their initial states by means of the software reset bit (SWR) in the EDMAC mode register (EDMR) in the EDMAC before making settings again.

PRM Bit (Promiscuous Mode)

Setting the PRM bit enables all Ethernet frames to be received. "All Ethernet frames" means all receivable frames, irrespective of differences or enabled/disabled status of destination address, broadcast address, multicast bit, etc.

RTM Bit (Transmission/Reception Rate)

This bit specifies the transmission and reception bit rate when RMII is selected.

TE Bit (Transmission Enable)

If a switch is made from transmitting function enabled (TE = 1) to disabled (TE = 0) while a frame is being transmitted, the transmitting function will be enabled until transmission of the corresponding frame is completed.

RE Bit (Reception Enable)

If a switch is made from receiving function enabled (RE = 1) to disabled (RE = 0) while a frame is being received, the receiving function will be enabled until reception of the corresponding frame is completed.

MPDE Bit (Magic Packet™ Detection Enable)

This bit enables or disables Magic Packet™ detection by hardware to allow activation from the Ethernet.

ZPF Bit (PAUSE Frame Usage with TIME = 0 Enable)

When the ZPF bit is set to 0, the next frame is not transmitted until the time specified by the Timer value has elapsed. On receiving a PAUSE frame with a Timer value of 0, the PAUSE frame is discarded.
When the ZPF bit is cleared to 0, if the data size in the receive FIFO becomes smaller than the setting of the flow control start FIFO threshold setting register (FCFTR) of the EDMAC before the time specified by the Timer value elapses, an automatic PAUSE frame with a Timer value of 0 is transmitted. On receiving a PAUSE frame with a Timer value of 0, the transmission wait state is canceled.

31.2.2 Receive Frame Length Register (RFLR)

Address(es): 000C 0108h

	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 Length 11 to 0	000h to 5EEh: 1,518 bytes 5EFh: 1,519 bytes 5F0h: 1,520 bytes : : 7FFh: 2,047 bytes 800h to FFFh: 2,048 bytes	R/W
b31 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RFLR specifies the maximum frame length (in bytes) that can be received by the RX63N. The RFLR settings must not be changed while the receiving function is enabled.

RFL[11:0] Bits (Receive Frame Length 11 to 0)

The RFL[11:0] bits specify a value for frame length to be checked. When the received data exceeds the specified value, a frame length error occurs.

The frame data described here refers to all fields from the destination address up to the CRC data, but actually, frame contents from the destination address up to the data are transferred to memory; CRC data is not included in the transfer. When the received data exceeds the specified length, the part of data that exceeds the specified length is discarded.

31.2.3 ETHERC Status Register (ECSR)

Address(es): 000C 0110h

	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	Illegal Carrier Detection	0: PHY-LSI has not detected an illegal carrier on the line 1: PHY-LSI has detected an illegal carrier on the line	R/W
b1	MPD	Magic Packet™ Detection	0: Magic Packet™ has not been detected 1: Magic Packet™ has been detected	R/W
b2	LCHNG	Link Signal Change	0: Change in the LINKSTA signal has not been detected 1: Change in the LINKSTA signal has been detected (high to low or low to high)	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b4	PSRTO	PAUSE Frame Retransmit Retry Over	0: PAUSE frame retransmit count has not exceeded the upper limit 1: PAUSE frame retransmit count has exceeded the upper limit	R/W
b5	BFR	Continuous Broadcast Frame Reception	0: Continuous reception of broadcast frames has not been detected. 1: Continuous reception of broadcast frames has been detected.	R/W
b31 to b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ECSR indicates the status in the ETHERC. Each state can be notified to the CPU by interrupts. When 1 is written to the BFR, PSRTO, LCHNG, MPD, and ICD bits, the corresponding flags can be cleared. Writing 0 does not affect the flags. For bits that generate interrupts, the interrupt can be enabled or disabled by the corresponding bit in the ETHERC interrupt permission register (ECSIPR).
The interrupts generated due to ECSR are indicated in the ETHERC status register source bit (ECI) in ETHERC/EDMAC status register (EESR) of the EDMAC.

ICD Bit (Illegal Carrier Detection)

This bit indicates that the PHY-LSI has detected an illegal carrier on the line.
The ICD bit is set to 1 when the signals transmitted from the PHY-LSI to the RX63N through the ET_RX_DV, ET_RX_ER, and ET_ERXD3 to ET_ERXD0 pins are 0, 1, and 1110, respectively (see Figure 31.9). If the signal input from the PHY-LSI changes in a period shorter than the period required for recognition by software, correct information may not be obtained. Refer to the timing specification of the PHY-LSI used.

LCHNG Bit (Link Signal Change)

This bit indicates that the ET_LINKSTA signal input from the PHY-LSI has changed from high to low or low to high. To check the current Link state, refer to the LNKSTA pin status bit (LMON) in the PHY status register (PSR).

PSRTO Bit (PAUSE Frame Retransmit Retry Over)

This bit indicates whether the retransmit count for retransmitting a PAUSE frame when flow control is enabled has exceeded the retransmit upper limit set in the automatic PAUSE frame retransmit count register (TPAUSER).

31.2.4 ETHERC Interrupt Permission Register (ECSIPR)

Address(es): 000C 0118h

	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	Illegal Carrier Detect Interrupt Enable	0: Interrupt notification by the ICD bit is disabled 1: Interrupt notification by the ICD bit is enabled	R/W
b1	MPDIP	Magic Packet™ Detect Interrupt Enable	0: Interrupt notification by the MPD bit is disabled 1: Interrupt notification by the MPD bit is enabled	R/W
b2	LCHNGIP	LINK Signal Change Interrupt Enable	0: Interrupt notification by the LCHNG bit is disabled 1: Interrupt notification by the LCHNG bit is enabled	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b4	PSRTOIP	PAUSE Frame Retransmit Retry Over Interrupt Enable	0: Interrupt notification by the PSRTO bit is disabled 1: Interrupt notification by the PSRTO bit is enabled	R/W
b5	BFSIPR	Continuous Broadcast Frame Reception Interrupt Enable	0: Interrupt notification by the corresponding bit in ECSR is disabled 1: Interrupt notification by the corresponding bit in ECSR is enabled	R/W
b31 to b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ECSIPR enables or disables the interrupt sources indicated by ECSR. Each bit can disable or enable interrupts corresponding to the bits in ECSR.

31.2.5 PHY Interface Register (PIR)

Address(es): 000C 0120h

	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	—	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MDC	MII/RMII Management Data Clock	The value set in the MDC bit is output from the ET_MDC pin to supply the MII/RMII with the management data clock.	R/W
b1	MMD	MII/RMII Management Mode	0: Read direction is specified 1: Write direction is specified	R/W
b2	MDO	MII/RMII Management Data-Out	Stores the data output from the ET_MDIO pin. The data is output from the ET_MDIO pin when the MMD bit is set to 1 (specifying the write direction). No data is output from the pin when the MMD bit is set to 0 (specifying the read direction)	R/W
b3	MDI	MII/RMII Management Data-In	Indicates the level of the ET_MDIO pin. The write value should always be 0.	R/W
b31 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PIR provides a means of accessing the PHY-LSI internal registers via the MII/RMII.

MDC Bit (MII/RMII Management Data Clock)

The value set in the MDC bit is output from the ET_MDC pin to supply the MII/RMII with the management data clock. For the method of accessing the MII/RMII registers, see section 31.3.4, Accessing MII/RMII Registers .

31.2.6 PHY Status Register (PSR)

Address(es): 000C 0128h

	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	—

Bit	Symbol	Bit Name	Description	R/W
b0	LMON	LINKSTA Pin Status	The Link status can be read by connecting the Link signal output from the PHY-LSI to the EX_LINKSTA pin. For the polarity, refer to the specifications of the PHY-LSI to be connected.	R
b31 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R

PSR is used to indicate interface signals from the PHY-LSI.

31.2.7 Random Number Generation Counter Upper Limit Setting Register (RDMLR)

Address(es): 000C 0140h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	RMD[19:0]			
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
	RMD[19:0]															
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]	Upper Limit for Counter Used in Random Number Generation Block	00000h: Setting for normal operation 00001h to FFFFEh: Setting for the upper limit for the counter used in the random number generation block	R/W
b31 to b20	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note: • The operation of the random number generation block depends on the setting in RDMLR. Accordingly, special attention should be paid when setting a value other than 0.

RDMLR specifies the upper limit for the counter used in the random number generation block.
This setting must not be changed while the transmitting and receiving functions of RDMLR are enabled.

31.2.8 IPG Register (IPGR)

Address(es): 000C 0150h

	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]	Inter Packet 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	These bits are always read as 0. The write value should always be 0.	R/W

IPGR sets the IPG (Inter Packet Gap) value. This setting must not be changed while the transmitting and receiving functions of ECMR are enabled. For details, refer to section 31.3.6, Operation by IPG Setting.

IPG[4:0] Bits (Inter Packet Gap)

These bits specify the IPG value in 4-bit time units.

31.2.9 Frame Register (APR)

Address(es): 000C 0154h

	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	These bits set the TIME parameter value of an automatic PAUSE frame.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

APR specifies the TIME parameter value of an automatic PAUSE frame. When an automatic PAUSE frame is transmitted, the value set in APR is used as the TIME parameter of the PAUSE frame.
This setting must not be changed while the transmitting and receiving functions of APR are enabled.

AP[15:0] Bits (Automatic PAUSE)

These bits set the TIME parameter value of an automatic PAUSE frame. One bit is equivalent to 512 bit-time.

31.2.10 Automatic PAUSE Manual PAUSE Frame Register (MPR)

Address(es): 000C 0158h

	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	These bits set the TIME parameter value of a manual PAUSE frame.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

MPR specifies the TIME parameter value of a manual PAUSE frame. When a manual PAUSE frame is transmitted, the value set in MPR is used as the TIME parameter of the PAUSE frame.
This setting can be changed while the transmitting and receiving functions of MPR are enabled.

MP[15:0] Bits (Manual PAUSE)

These bits set the TIME parameter value of a manual PAUSE frame. One bit is equivalent to 512 bit-time. The read value is undefined.

31.2.11 PAUSE Frame Receive Counter Register (RFCF)

Address(es): 000C 0160h

	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]	PAUSE Frame Receive Count	Receive counter	R
b31 to b8	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RFCF is a counter that indicates the number of times a PAUSE frame was received.

31.2.12 Retransmit Count Register (TPAUSER)

Address(es): 000C 0164h

	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]	Upper Limit for Automatic PAUSE Frame Retransmission	0000h: Retransmit count is unlimited 0001h: Retransmit count is 1 : FFFFh: Retransmit count is 65,535	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TPAUSER specifies the upper limit for the number of times to retransmit an automatic PAUSE frame. The settings in TPAUSER must not be changed while the transmitting function is enabled.

31.2.13 Automatic PAUSE PAUSE Frame Retransmit Counter Register (TPAUSECR)

Address(es): 000C 0168h

	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	These bits are always read as 0. The write value should always be 0.	R/W

TPAUSECR is a counter that indicates the number of times a PAUSE frame was retransmitted.

31.2.14 Broadcast Frame Receive Count Setting Register (BCFRR)

Address(es): 000C 016Ch

	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]	Receive Count for Continuous Broadcast Frames	0000h: No limitation for receive count 0001h: 1 frame can be received : : FFFFh: 65,535 frames can be received	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

BCFRR specifies the number of Broadcast frames that can be received continuously. This setting must not be changed while the transmitting and receiving functions of BCFRR are enabled.

BCF[15:0] Bits (Receive Count for Continuous Broadcast Frames)

If the destination address (DA) can receive a frame with a Broadcast address up to the number of times set in these bits and frames have been received for more times than the specified count, the excess Broadcast frames are discarded.

31.2.15 MAC Address High Register (MAHR)

Address(es): 000C 01C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MA[47:16]															
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
	MA[47:16]															
Value after 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	MA[47:16]	MAC Address Bits 47 to 16	These bits specify the upper 32 bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 01234567h in MAHR.	R/W

MAHR specifies the upper 32 bits of the 48-bit MAC address.
The settings in MAHR should be made in the initialization process after a reset.
The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the ETHERC and EDMAC to their initial states by means of the software reset bit (SWR) in the EDMAC mode register (EDMR) in the EDMAC before making settings again.

31.2.16 Frame MAC Address Low Register (MALR)

Address(es): 000C 01C8h

	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
	MA[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	MA[15:0]	MAC Address Bits 15 to 0	These bits specify the lower 16 bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 89ABh in MALR.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

MALR specifies the lower 16 bits of the 48-bit MAC address.
The settings in MALR should be made in the initialization process after a reset.
The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the ETHERC and EDMAC to their initial states by means of the software reset bit (SWR) in the EDMAC mode register (EDMR) in the EDMAC before making settings again.

31.2.17 Transmit Retry Over Counter Register (TROCR)

Address(es): 000C 01D0h

	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 number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer.	R/W

TROCR is a counter that indicates the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer. When 16 transmission attempts have failed, TROCR is incremented by 1. When the value in TROCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to TROCR with any value.

31.2.18 Delayed Collision Detect Counter Register (CDCR)

Address(es): 000C 01D4h

	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 number of all delayed collisions after the start of data transmission.	R/W

CDCR is a counter that indicates the number of all delayed collisions that occurred on the line after the start of data transmission. When the value in CDCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to CDCR with any value.

31.2.19 Lost Carrier Counter Register (LCCR)

Address(es): 000C 01D8h

	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 number of times the carrier was lost during data transmission.	R/W

LCCR is a counter that indicates the number of times the carrier was lost during data transmission. When the value in LCCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to LCCR with any value.

31.2.20 Carrier Not Detect Counter Register (CNDCR)

Address(es): 000C 01DCh

	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 number of times carrier was not detected.	R/W

CNDCR is a counter that indicates the number of times carrier was not detected during preamble transmission. When the value in CNDCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to CNDCR with any value.

31.2.21 CRC Error Frame Receive Counter Register (CEFCR)

Address(es): 000C 01E4h

	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 number of CRC error frames received.	R/W

CEFCR is a counter that indicates the number of times a frame with a CRC error was received. When the value in CEFCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to CEFCR with any value.

31.2.22 Frame Receive Error Counter Register (FRECR)

Address(es): 000C 01E8h

	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 number of errors during frame reception.	R/W

FRECR is a counter that indicates the number of frames for which a receive error was generated by the signal input to the ET_RX_ER pin from the PHY-LSI. FRECR is incremented each time the ET_RX_ER signal becomes active. When the value in FRECR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to FRECR with any value.

31.2.23 Too-Short Frame Receive Counter Register (TSFRCR)

Address(es): 000C 01ECh

	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 number of frames received with a length of less than 64 bytes.	R/W

TSFRCR is a counter that indicates the number of frames received with a length less than 64 bytes. When the value in TSFRCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to TSFRCR with any value.

31.2.24 Too-Long Frame Receive Counter Register (TLFRCR)

Address(es): 000C 01F0h

	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 number of frames received with a length exceeding the value in RFLR.	R/W

TLFRCR is a counter that indicates the number of frames received with a length exceeding the value specified by RFLR. When the value in TLFRCR reaches FFFFFFFFh, the counter stops incrementing. TLFRCR is not incremented when a frame containing residual bits is received. In this case, the reception of the frame is indicated in the residual-bit frame receive counter register (RFCR). The counter value is cleared to 0 by a write to TLFRCR with any value.

31.2.25 Residual-Bit Frame Receive Counter Register (RFCR)

Address(es): 000C 01F4h

	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 number of frames received containing residual bits.	R/W

RFCR is a counter that indicates the number of frames received containing residual bits (less than an 8-bit unit). When the value in RFCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to RFCR with any value.

31.2.26 Multicast Address Frame Receive Counter Register (MAFCR)

Address(es): 000C 01F8h

	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 number of multicast frames received.	R/W

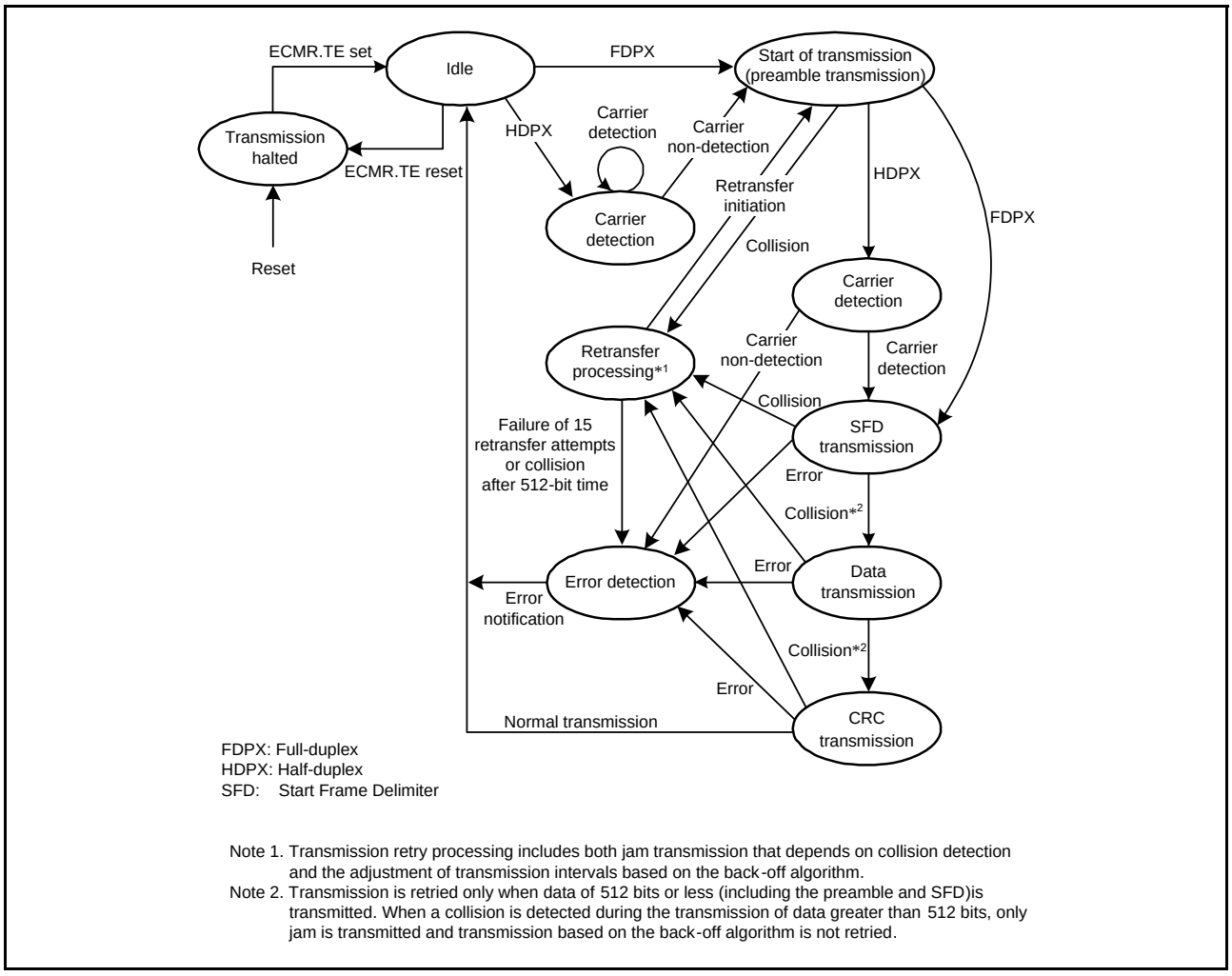
MAFCR is a counter that indicates the number of frames received with a specified multicast address. When the value in MAFCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to MAFCR with any value.

31.3 Operation

The following gives an outline of the ETHERC operations.
The ETHERC supports flow control functions conforming to IEEE802.3x, and can transmit and receive PAUSE frames used for the control.

31.3.1 Transmission

The ETHERC transmitter assembles transmit data into a frame and outputs it to the MII/RMII when there is a transmit request from the transmit EDMAC. The data transmitted via the MII/RMII is transmitted to the lines by the PHY-LSI. Figure 31.2 shows the state transitions in the ETHERC transmitter.



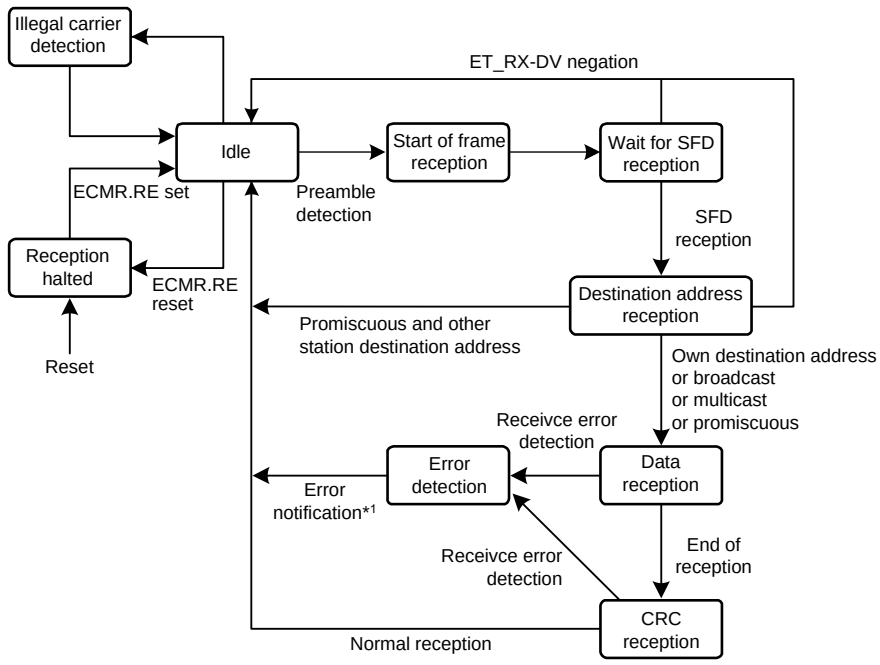
Note 1. Transmission retry processing includes both jam transmission that depends on collision detection and the adjustment of transmission intervals based on the back-off algorithm.
Note 2. Transmission is retried only when data of 512 bits or less (including the preamble and SFD)is transmitted. When a collision is detected during the transmission of data greater than 512 bits, only jam is transmitted and transmission based on the back-off algorithm is not retried.

Figure 31.2 ETHERC Transmitter State Transitions

1. When the transmit enable (ECMR.TE) bit is set, the transmitter enters the transmit idle state.
2. When a transmit request is issued by the transmit EDMAC, the ETHERC sends the preamble to the MII/RMII after carrier detection and a transmission delay equivalent to the frame interval time. If full-duplex transfer is selected, which does not require carrier detection, the preamble is sent as soon as a transmit request is issued by the transmit EDMAC.
3. The transmitter sends the SFD, data, and CRC sequentially. At the end of transmission, the transmit EDMAC generates a transmission complete interrupt (TC). If a collision or the carrier-not-detected state occurs during data transmission, it is reported as an interrupt.
4. After waiting for the frame interval time, the transmitter enters the idle state, and if there is more transmit data, continues transmission.

31.3.2 Reception

The ETHERC receiver separates the frame from the MII/RMII into preamble, SFD, data, and CRC, and the fields from DA (destination address) to the CRC data are transferred to the receive EDMAC. Figure 31.3 shows the state transitions of the ETHERC receiver.



SFD: Start frame delimiter

Note 1. The error frame also transmits data to the buffer.

Figure 31.3 ETHERC Receiver State Transitions

1. When the receive enable (ECMR.RE) bit is set, the receiver enters the receive idle state.
2. Upon detecting an SFD (start frame delimiter) after a receive packet preamble, the receiver starts receive processing. It discards a frame with an invalid pattern.
3. In normal mode, if the destination address of the frame matches the RX63N address, or if the broadcast or multicast frame type is specified, the receiver starts data reception. In promiscuous mode, the receiver starts reception for any type of frame.
4. After data reception from the MII/RMII, the receiver carries out a CRC check in the frame data field. The result is indicated as a status bit in the descriptor after the frame data has been written to memory. The receiver reports an error status in the case of an abnormality.
5. After one frame has been received, if the receive enable bit is set (ECMR.RE = 1) in the ETHERC mode register, the receiver prepares to receive the next frame.

31.3.3 Frame Timing

31.3.3.1 MII Frame Timing

The MII Frame timing is shown in Figure 31.4 to Figure 31.9.

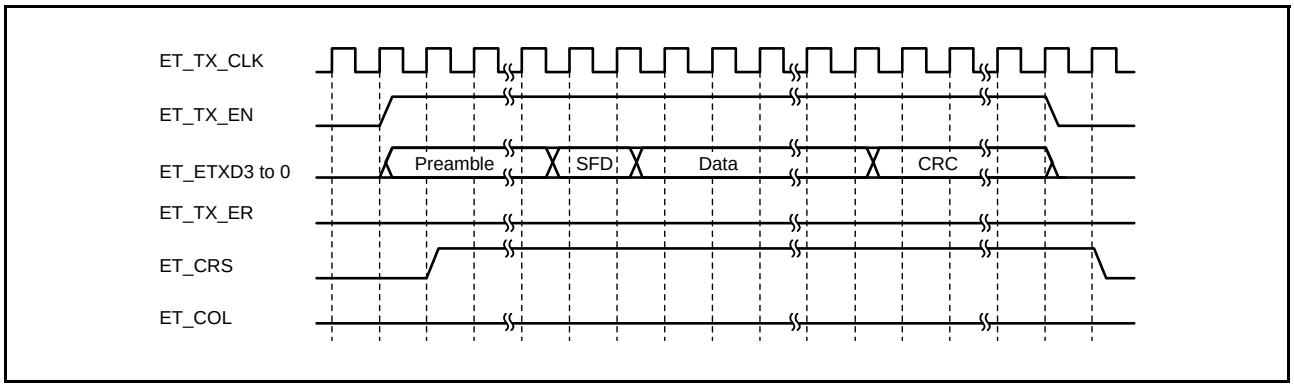


Figure 31.4 MII Frame Transmit Timing (Normal Transmission)

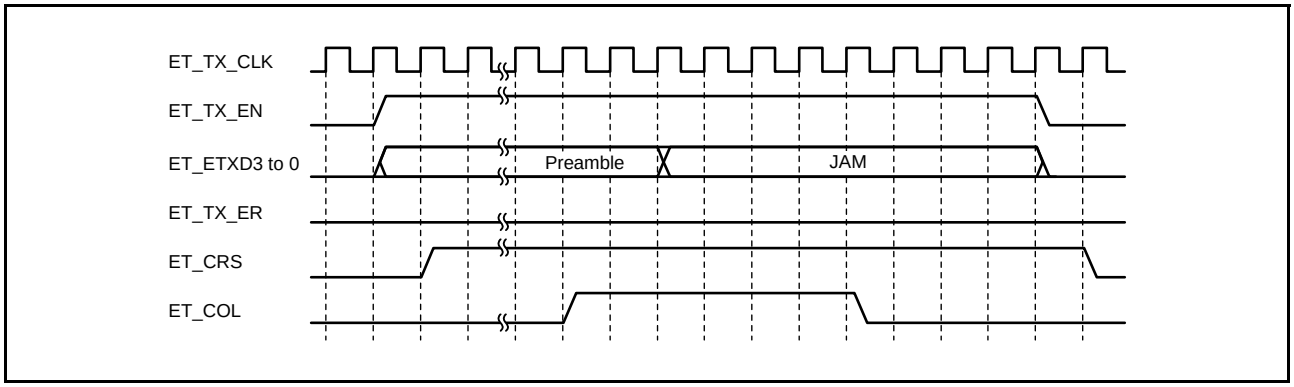


Figure 31.5 MII Frame Transmit Timing (Collision)

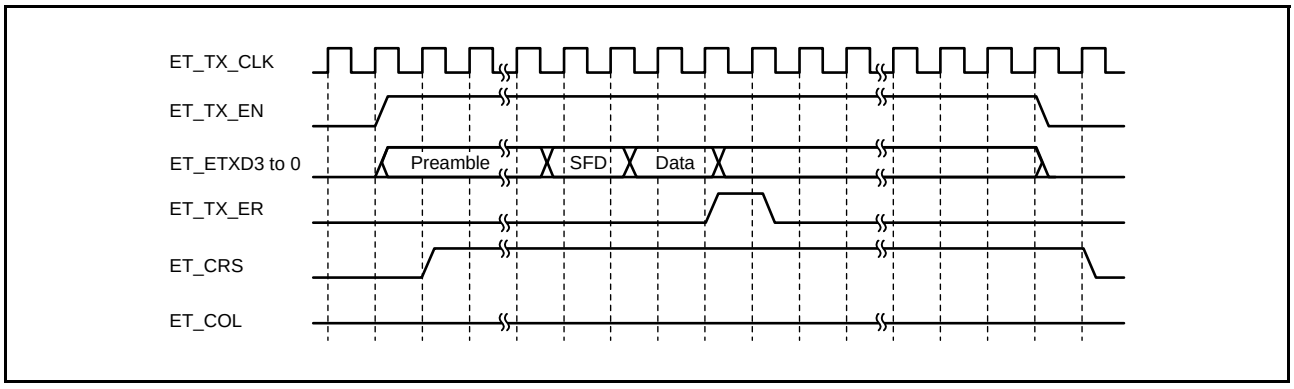


Figure 31.6 MII Frame Transmit Timing (Transmit Error)

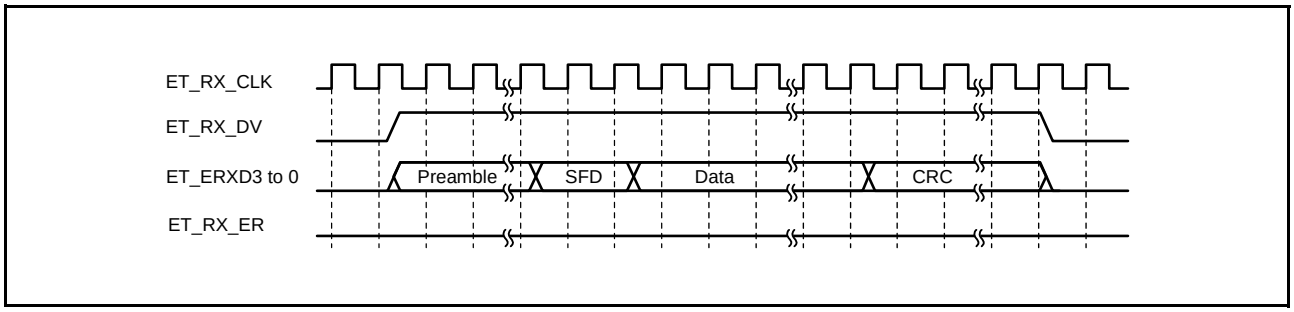


Figure 31.7 MII Frame Receive Timing (Normal Reception)

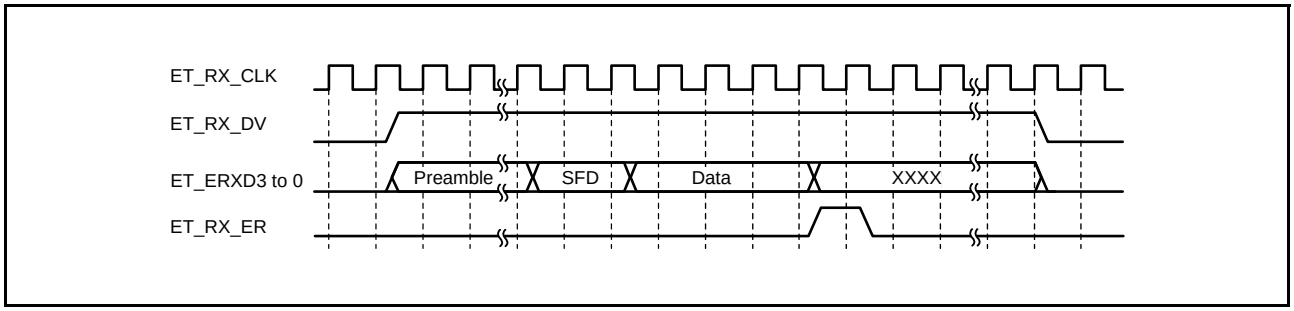


Figure 31.8 MII Frame Receive Timing (Reception Error (1): Receive Error Notification)

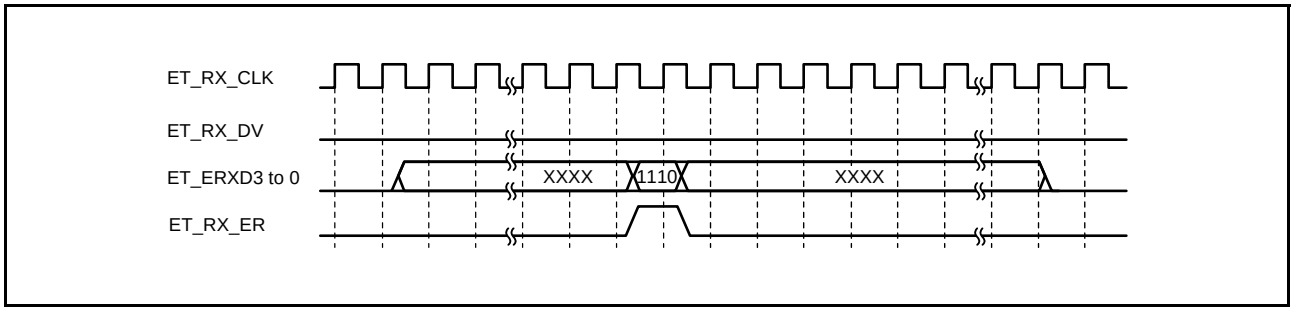


Figure 31.9 MII Fame Receive Timing (Reception Error (2): Carrier Error Notification)

31.3.3.2 RMII Frame Timing

The RMII Frame timing is shown in Figure 31.10 to Figure 31.12.

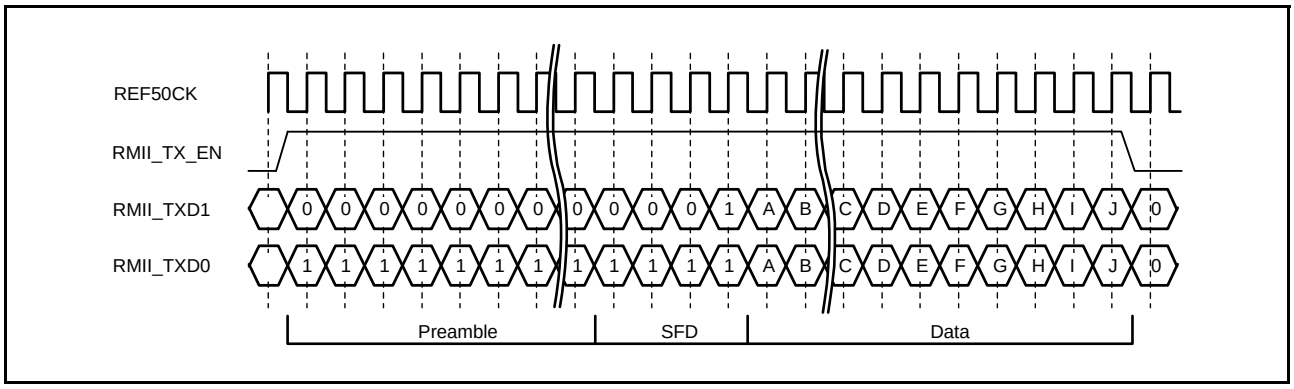


Figure 31.10 RMII Frame Transmit Timing (Normal Transmission)

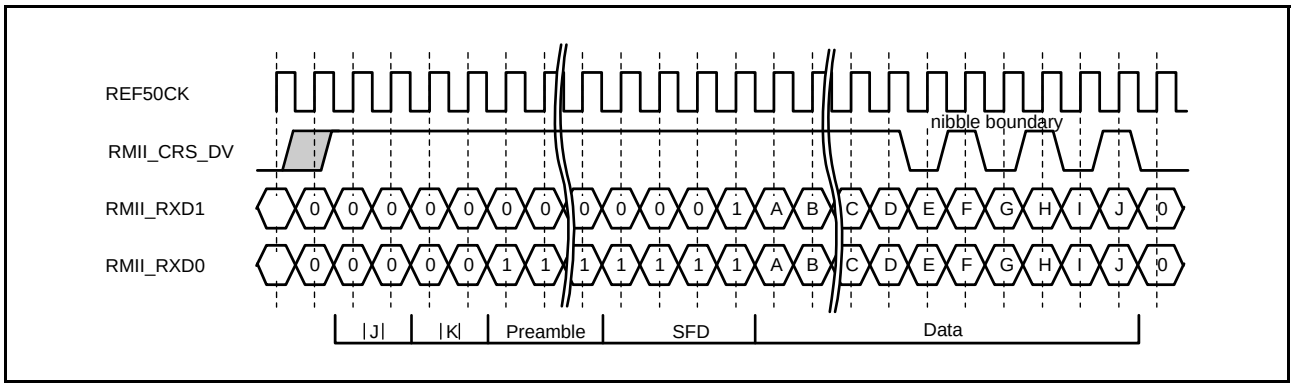


Figure 31.11 RMII Frame Receive Timing (Normal Reception)

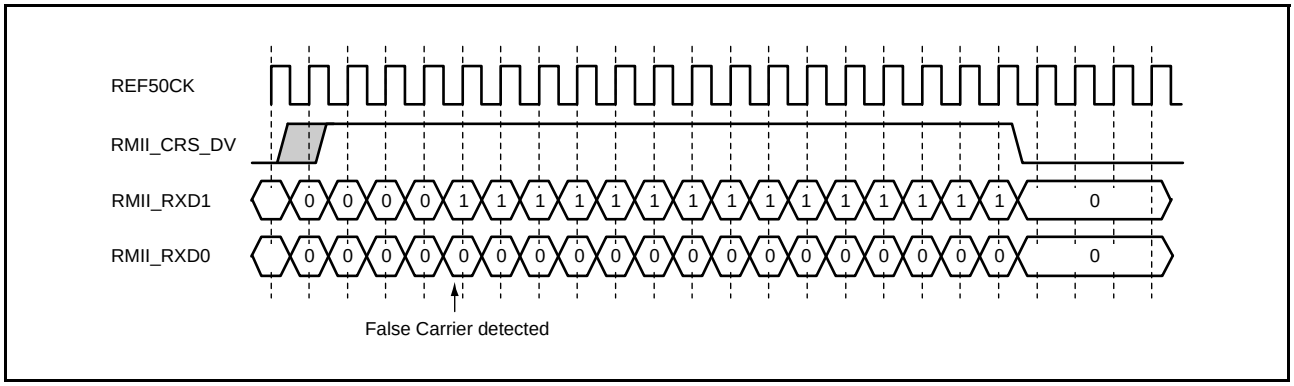


Figure 31.12 RMII Frame Receive Timing (False Carrier Detected)

31.3.4 Accessing MII/RMII Registers

The MII/RMII registers in the PHY-LSI are accessed via the PHY interface register (PIR) in the RX63N. Connection is made as a serial interface in accordance with the MII/RMII frame format.

31.3.4.1 MII/RMII Management Frame Format

The format of an MII/RMII management frame is shown in Figure 31.13. To access an MII/RMII register, a management frame should be implemented by the program in accordance with the procedures shown in section 31.3.4.2, MII/RMII Register Access Procedure.

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	—
Read	1..1	01	10	00001	RRRRR	Z0	D..D	—
Write	1..1	01	01	00001	RRRRR	10	D..D	X

PRE: 32 consecutive 1b
ST: Write 01b to indicate the start of frame
OP: Write the code indicating access type
PHYAD: Write 0001b if the PHY-LSI address is 1 (sequential write starting with the MSB).
This value depends on the PHY-LSI address.
REGAD: Write 0001b if the register address is 1 (sequential write starting with the MSB).
This value depends on the PHY-LSI register address.
TA: Time for switching data transmission source on MII/RMII interface
(a) For write: Write 10b
(b) For read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequentially write or read from MSB
(a) For write: Write 16-bit data
(b) For read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) For write: Independent bus release (notation: X) performed
(b) For read: Bus already released in TA; control unnecessary

Figure 31.13 MII/RMII Management Frame Format

31.3.4.2 MII/RMII Register Access Procedure

The program should access MII/RMII registers via the PHY interface register (PIR). Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. Figure 31.14 to Figure 31.17 show the MII/RMII register access timing. The timing will differ depending on the PHY-LSI type.

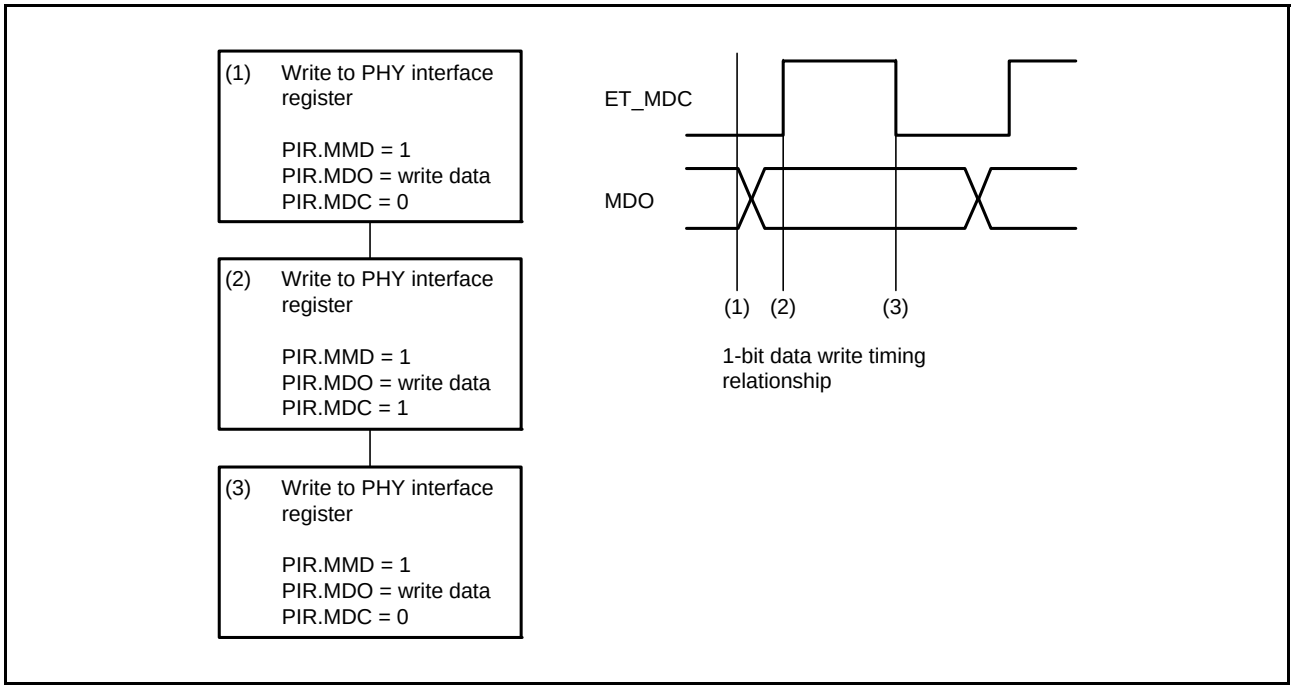


Figure 31.14 Flowchart of 1-Bit Data Write

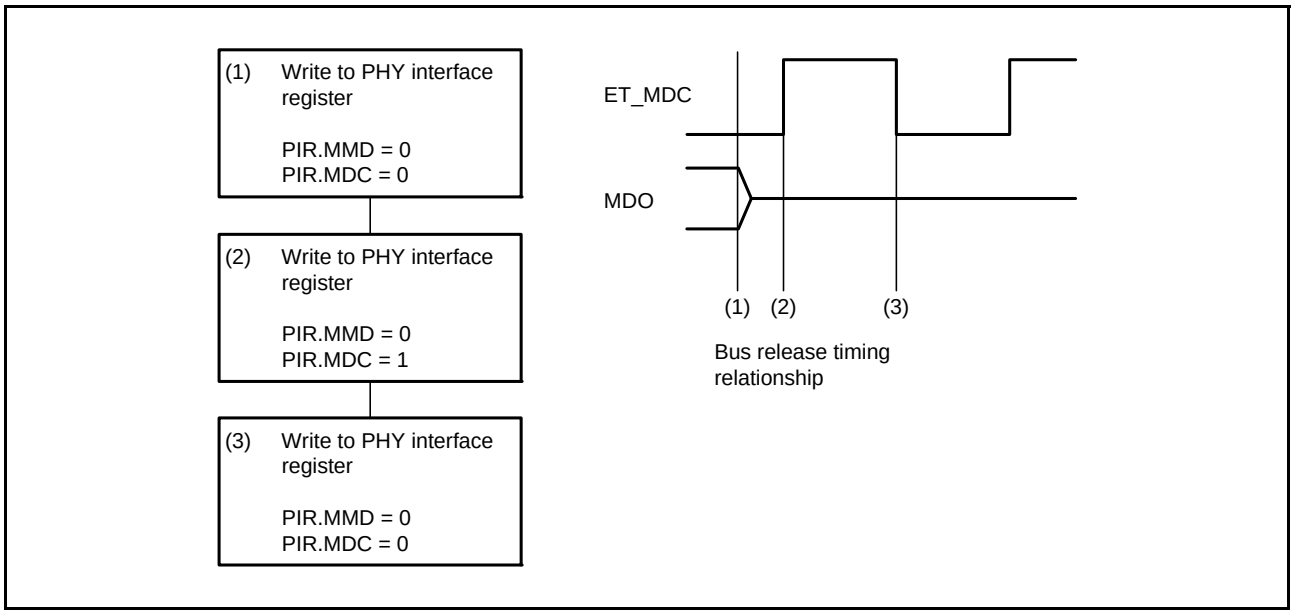


Figure 31.15 Flowchart of Bus Release (TA in Read in Figure 31.13)

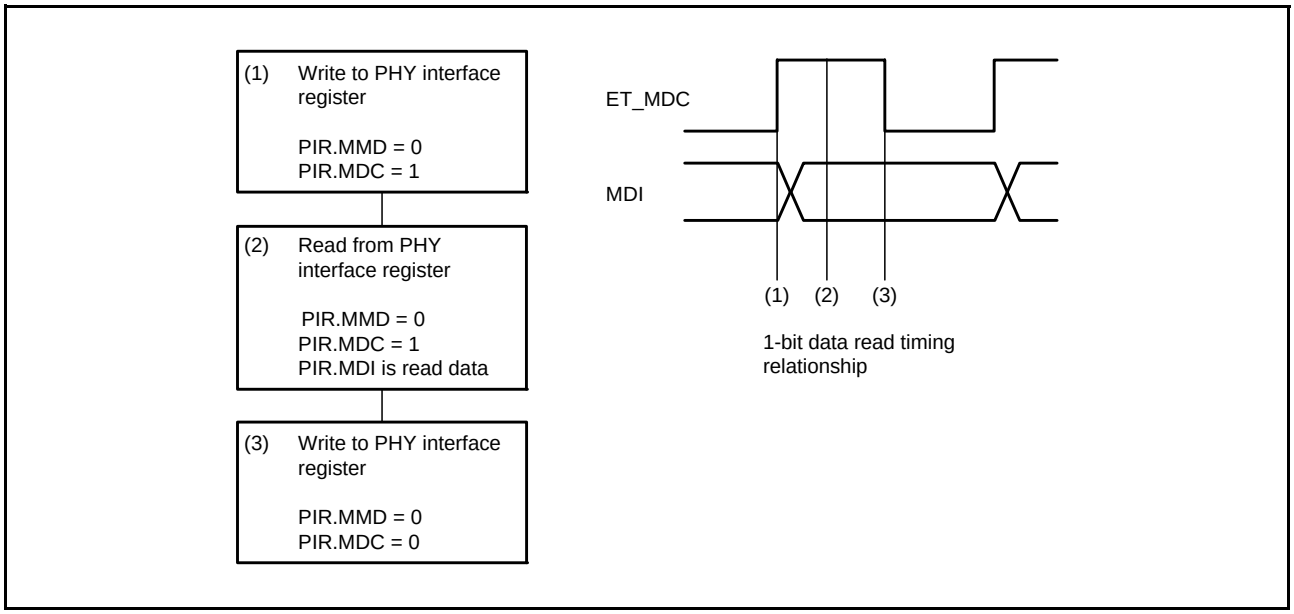


Figure 31.16 Flowchart of 1-Bit Data Read

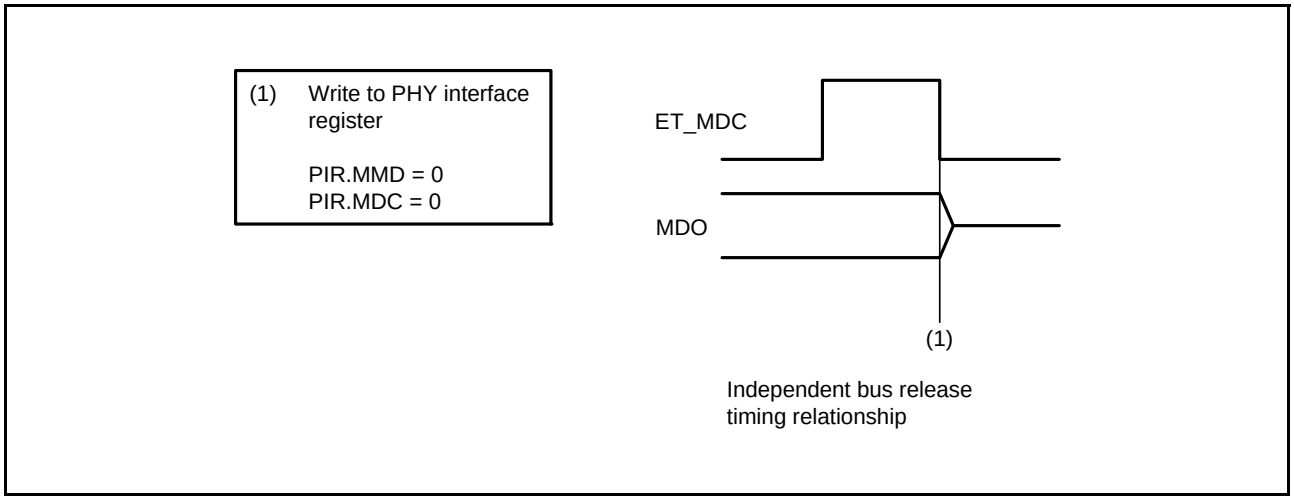


Figure 31.17 Flowchart of Independent Bus Release (IDLE in Write in Figure 31.13)

31.3.5 Magic Packet™ Detection

The ETHERC has a Magic Packet™ detection function. This function provides a Wake-On-LAN (WOL) facility that activates various peripheral devices connected to a LAN from the host device or other source. This makes it possible to construct a system in which a peripheral device receives a Magic Packet™ sent from the host device or other source, and activates itself. When the Magic Packet™ is detected, data is stored in the receive FIFO through the broadcast packet that has been received and the ETHERC is notified of the receiving status. To return to normal operation from the interrupt processing, initialize the ETHERC and EDMAC by using the software reset (SWR) bit in the EDMAC mode register (EDMR) in the EDMAC.

With a Magic Packet™, reception is performed regardless of the destination address. As a result, this function becomes valid and the ET_WOL pin is enabled only when a match occurs with the destination address specified by the format in the Magic Packet™. Further information on Magic Packets™ can be found in the technical documentation published by Advanced Micro Devices, Inc.

The procedure for using the WOL function with the RX63N is as follows.

1. Disable interrupt source output by means of the interrupt enable/mask registers.
2. Set the Magic Packet™ detection enable bit (MPDE) in the ETHERC mode register (ECMR).
3. Set the Magic Packet™ detection interrupt enable bit (MPDIP) to the enabled state in the ETHERC interrupt permission register (ECSIPR).
4. If necessary, set the CPU operating mode to sleep mode or set peripheral modules to module standby mode.
5. When a Magic Packet™ is detected, an interrupt is sent to the CPU. The ET_WOL pin notifies peripheral LSIs that the Magic Packet™ has been detected.

31.3.6 Operation by IPG Setting

The ETHERC has a function to change the non-transmission period, IPG (Inter Packet Gap), between transmit frames. By changing the value in the IPG setting register (IPGR), the transmission efficiency can be raised or lowered from the standard value. IPG settings are prescribed in the IEEE802.3 standard. When changing settings, adequately check that each device can operate smoothly on the same network.

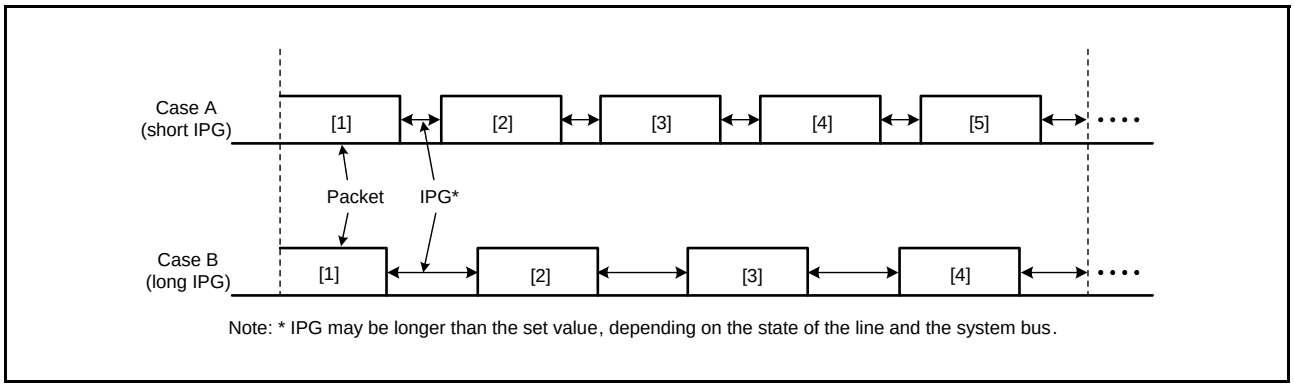


Figure 31.18 Relationship between IPG Setting and Transmission Efficiency

31.3.7 Flow Control

The ETHERC supports flow control functions conforming to IEEE802.3x for full-duplex operation. The flow control can be applied to both receive and transmit operations. For flow control, the following procedures can be used to transmit PAUSE frames.

31.3.7.1 Automatic PAUSE Frame Transmission

For receive frames, PAUSE frames are automatically transmitted when the amount of data written to the receive FIFO (in EDMAC) reaches the value set in the flow control start FIFO threshold setting register (FCFTR) of the EDMAC. The TIME parameter included in the PAUSE frame is set by the automatic PAUSE frame register (APR). The automatic PAUSE frame transmission is repeated until the amount of data in the receive FIFO becomes less than the value set in FCFTR as the receive data is read from the FIFO. Using the automatic PAUSE frame retransmit count register (TPAUSER), the upper limit of the PAUSE frame retransmission count can also be set. In this case, PAUSE frame transmission is repeated until the amount of receive FIFO data becomes less than the FCFTR value or the number of transmits reaches the TPAUSER value.
The automatic PAUSE frame transmission is enabled when the operating mode bit for transmitting port flow control (TXF) in the ETHERC mode register (ECMR) is 1.

31.3.7.2 Manual PAUSE Frame Transmission

PAUSE frames are transmitted under software control. When the Timer value is written to the manual PAUSE frame register (MPR), manual PAUSE frame transmission is started. With this method, PAUSE frame transmission is carried out only once.

31.3.7.3 PAUSE Frame Reception

After a PAUSE frame is received, the next frame is not transmitted until the time indicated by the Timer value elapses. However, the transmission of the current frame is continued. PAUSE frame reception is enabled when the operating mode bit for receiving port flow control (RXF) in the ETHERC mode register (ECMR) is set to 1.

31.4 Connection to PHY-LSI

Figure 31.19 and Figure 31.20 show examples of connection to the PHY-LSI.

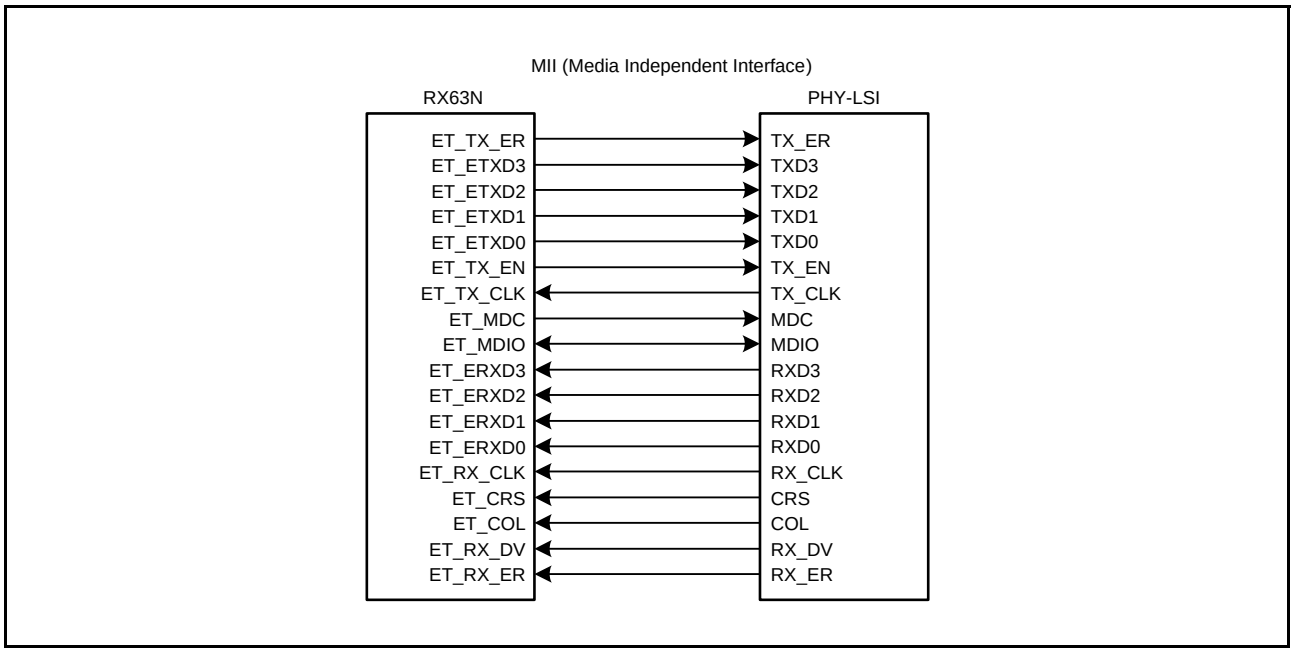


Figure 31.19 Example of Connection to PHY-LSI (MII)

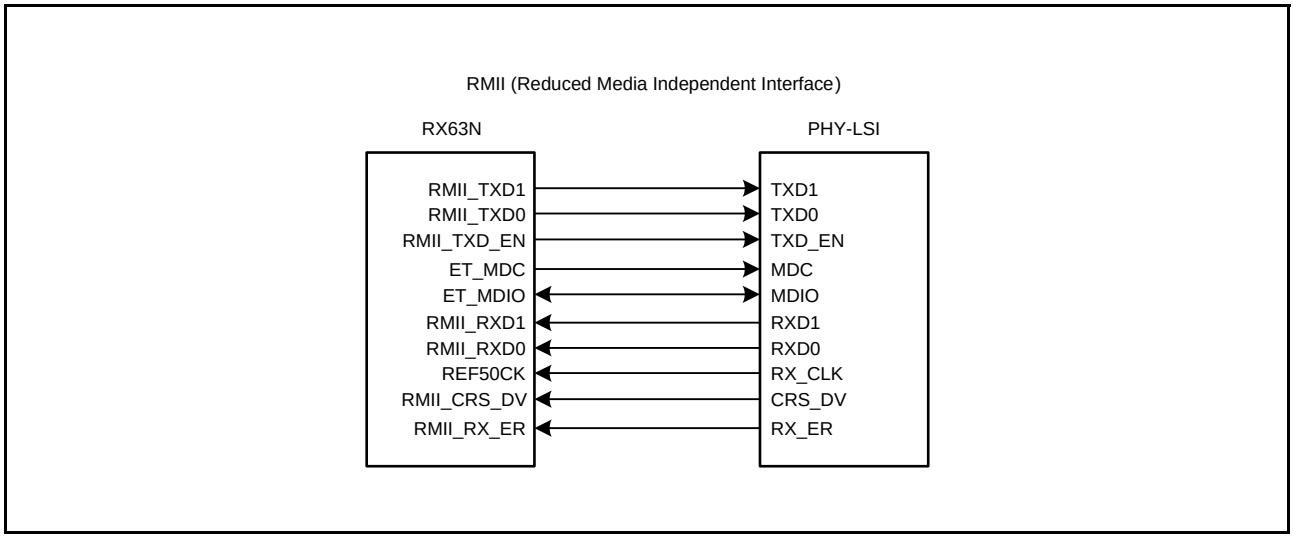


Figure 31.20 Example of Connection to PHY-LSI (RMII)

31.5 Usage Notes

Note the following when using the ETHERC.

31.5.1 Conditions for Setting LCHNG Bit

The ECSR.LCHNG bit may be set even when the input level on the ET_LINKSTA pin has not changed. It may be set when the ET_LINKSTA pin is selected by the PSEL bit in the GPIO or when a high level is applied to the ET_LINKSTA pin while the ETHERC/EDMAC is released from the software reset state by the SWR bit in EDMR in the EDMAC. This is because the ET_LINKSTA signal is internally fixed low regardless of the external pin level when the ET_LINKSTA pin is not selected by the GPIO or while the ETHERC/EDMAC is in the software reset state,.
In order not to request the LINK signal change interrupt accidentally, clear the ECSR.LCHNG bit before setting the ECSIPR.LCHNGIP.

31.5.2 Input on the RMII_RX_ER Pin While the RMII is Selected

When the RMII is selected, if the width of a reception error signal picked up from the PHY layer is only equivalent to one cycle of the 50-MHz reference clock for the RMII, the signal will not be recognized as an error signal.

32. Ethernet Controller Direct Memory Access Controller (EDMAC)

32.1 Overview

The RX63N Group has an on-chip direct memory access controller (EDMAC) directly connected to the Ethernet controller (ETHERC). The EDMAC controls the most part of the buffer management by using descriptors. This reduces the load on the CPU, thus enabling efficient data transmission and reception.
Table 32.1 shows the EDMAC specifications, and Figure 32.1 shows the configuration of the EDMAC, and the descriptors and transmit/receive buffers in memory.

Table 32.1 Specifications of EDMAC

Item	Description
Data transmission and reception	• Descriptor management system • Supports single-frame/multi-buffer operation
Function	• Achieves efficient system bus utilization through the use of DMA block transfer (32-byte units) • Transmit/receive frame status information is indicated in descriptors • Padding can be inserted in receive data

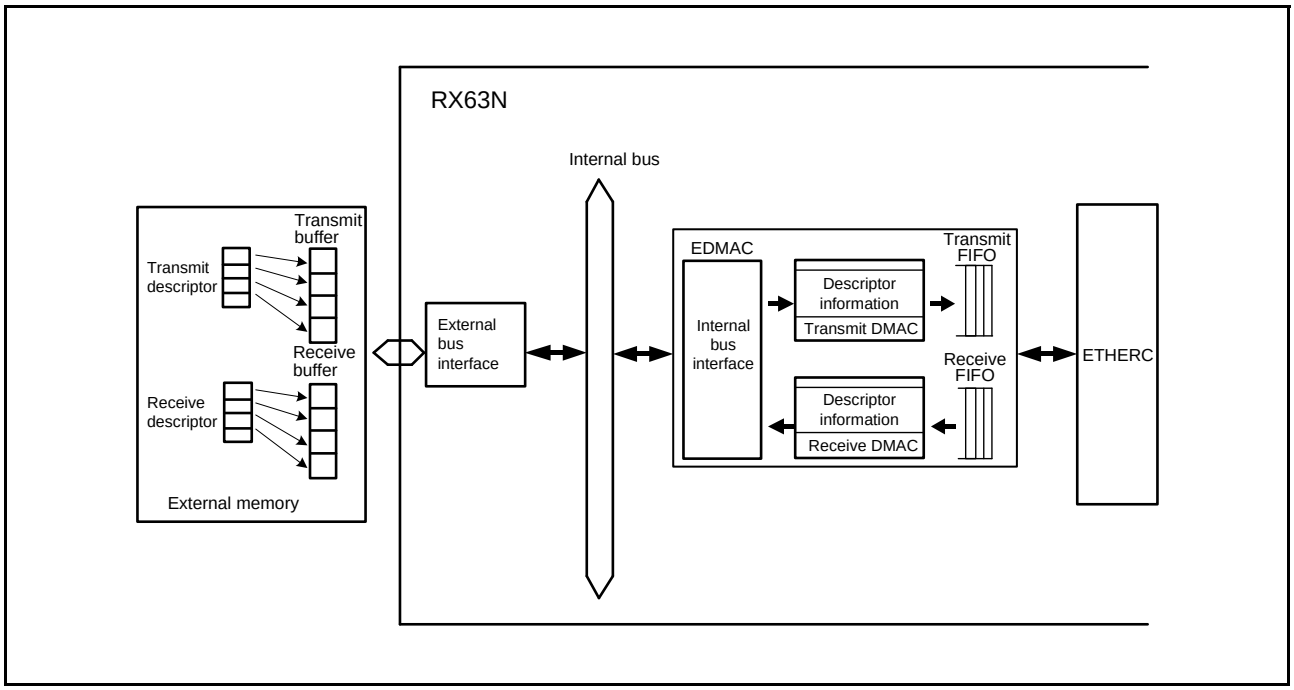


Figure 32.1 Configuration of EDMAC, and Descriptors and Buffers

32.2 Register Descriptions

32.2.1 EDMAC Mode Register (EDMR)

Address(es): 000C 0000h

	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	[Writing] 0: Disabled 1: Internal hardware is reset*1 [Reading] Always read as 0.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should always 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*2	0: Big endian mode (longword access) 1: Little endian mode (longword access)	R/W
b31 to b7	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. Registers other than TDLAR, RMFCR, TFUCR, and RFOCR are reset.
Note 2. This setting is effective for received data and data for transmission. It does not apply to descriptors or registers (support is only for big endian).

EDMR specifies EDMAC operating mode.
EDMR should usually be set at initialization after a reset. If the ETHERC and EDMAC are initialized with EDMR during data transmission, abnormal data may be transmitted on the line.
It is prohibited to modify the operating mode while transmission or reception function is enabled. Before modifying the operating mode, the ETHERC and EDMAC should be initialized by setting the SWR bit.
Note that it takes 64 cycles of EDMAC internal bus clock for the ETHERC and EDMAC to be completely initialized. Therefore, the registers in the ETHERC or EDMAC should be accessed after that.

32.2.2 EDMAC Transmit Request Register (EDTRR)

Address(es): 000C 0008h

	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	0: Transmission-halted state Writing 0 does not stop transmission. Termination of transmission is controlled by the active bit of the transmit descriptor. 1: Transmission start The EDMAC starts reading the target descriptor and sends a frame whose transmission active bit is set to 1.	R/W
b31 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

EDTRR issues transmit directives to the EDMAC.
After having transmitted one frame, the EDMAC reads the next descriptor. If the transmit descriptor active bit in this descriptor is set (active), the EDMAC continues transmission. Otherwise, the EDMAC clears the TR bit and stops the transmit DMAC operation.

32.2.3 EDMAC Receive Request Register (EDRRR)

Address(es): 000C 0010h

	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: Receiving function is disabled*1 1: Receive descriptor is read, and the EDMAC becomes ready to receive	R/W
b31 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. If the receiving function is disabled during frame reception, write-back is not performed successfully to the receive descriptor. Following pointers to read a receive descriptor become abnormal and the EDMAC cannot operate successfully. In this case, to make EDMAC reception enabled again, execute a software reset by the EDMR.SWR bit.
To disable the EDMAC receiving function without executing a software reset, specify the ECMR.RE bit in the ETHERC. Next, after the EDMAC has completed the reception and write-back to the receive descriptor has been confirmed, disable the receiving function using EDRRR.

EDRRR issues receive directives to the EDMAC.
After writing 1 to the RR bit in this register, the EDMAC reads the receive descriptor. If the receive descriptor active bit of this receive descriptor is set to 1 (active), the EDMAC becomes ready for a receive request from the ETHERC.
After data has been received for the receive buffer size, the EDMAC reads the next receive descriptor and becomes ready for frame reception. If the receive descriptor active bit of that receive descriptor is set to 0 (inactive), the EDMAC clears the RR bit and stops receive DMAC operation.

32.2.4 Transmit Descriptor List Start Address Register (TDLAR)

Address(es): 000C 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDLA[31:0]															
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
	TDLA[31:0]															
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
b31 to b0	TDLA[31:0]	Transmit Descriptor Start Address	16-byte boundary: TDLA[3:0] = 0000b 32-byte boundary: TDLA[4:0] = 00000b 64-byte boundary: TDLA[5:0] = 000000b	R/W

TDLAR specifies the start address of the transmit descriptor list.
Descriptors have a boundary configuration in accordance with the descriptor length indicated by the EDMR.DL[1:0] bits.
TDLAR must not be modified during transmission. Modifications to TDLAR should only be made in the transmission-halted state specified by the EDTRR.TR bit (= 0).

TDLA[31:0] Bits (Transmit Descriptor Start Address)

The lower bits should be set according to the specified descriptor length.

32.2.5 Receive Descriptor List Start Address Register (RDLAR)

Address(es): 000C 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDLA[31:0]															
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
	RDLA[31:0]															
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
b31 to b0	RDLA[31:0]	Receive Descriptor Start Address	16-byte boundary: RDLA[3:0] = 0000b 32-byte boundary: RDLA[4:0] = 00000b 64-byte boundary: RDLA[5:0] = 000000b	R/W

RDLAR specifies the start address of the receive descriptor list.
Descriptors have a boundary configuration in accordance with the descriptor length indicated by the EDMR.DL[1:0] bits.
RDLAR must not be modified during reception. Modifications to RDLAR should only be made while reception is disabled by the EDRRR.RR bit (= 0).

RDLA[31:0] Bits (Receive Descriptor Start Address)

The lower bits should be set according to the specified descriptor length.

32.2.6 ETHERC/EDMAC Status Register (EESR)

Address(es): 000C 0028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWB	—	—	—	TABT	RABT	RFCOF	ADE	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 on Received Frame	0: CRC error has not been detected 1: CRC error has been detected	R/W
b1	PRE	PHY-LSI Receive Error	0: PHY-LSI receive error has not been detected 1: PHY-LSI receive error has been detected	R/W
b2	RTSF	Receive Too-Short Frame	0: Too-short frame has not been received 1: Too-short frame has been received	R/W
b3	RTLF	Receive Too-Long Frame	0: Too-long frame has not been received 1: Too-long frame has been received	R/W
b4	RRF	Receive Residual-Bit Frame	0: Residual-bit frame has not been received 1: Residual-bit frame has been received	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	RMAF	Receive Multicast Address Frame	0: Multicast address frame has not been received 1: Multicast address frame has been received	R/W
b8	TRO	Transmit Retry Over	0: Transmit retry-over condition has not been detected 1: Transmit retry-over condition has been detected	R/W
b9	CD	Delayed Collision Detect	0: Delayed collision has not been detected 1: Delayed collision has been detected	R/W
b10	DLC	Detect Loss of Carrier	0: Loss of carrier has not been detected 1: Loss of carrier has been detected	R/W
b11	CND	Carrier Not Detect	0: A carrier is detected when transmission starts 1: A carrier is not detected	R/W
b15 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b16	RFOF	Receive FIFO Overflow	0: Overflow has not occurred 1: Overflow has occurred	R/W
b17	RDE	Receive Descriptor Empty	0: Receive descriptor active bit RD0.RACT = 1 detected 1: Receive descriptor active bit RD0.RACT = 0 detected	R/W
b18	FR	Frame Reception	0: Frame has not been received 1: Frame has been received	R/W
b19	TFUF	Transmit FIFO Underflow	0: Underflow has not occurred 1: Underflow has occurred	R/W
b20	TDE	Transmit Descriptor Empty	0: Transmit descriptor active bit TD0.TACT = 1 detected 1: Transmit descriptor active bit TD0.TACT = 0 detected	R/W
b21	TC	Frame Transmit Complete	0: Transfer not complete, or no transfer directive issued 1: Transfer complete	R/W
b22	ECI	ETHERC Status Register Source	0: ETHERC status interrupt source has not been detected 1: ETHERC status interrupt source has been detected	R*1
b23	ADE	Address Error	0: Illegal memory address has not been detected (normal operation) 1: Illegal memory address has been detected*2	R/W

Bit	Symbol	Bit Name	Description	R/W
b24	RFCOF	Receive Frame Counter Overflow	0: Receive frame counter has not overflowed 1: Receive frame counter has overflowed	R/W
b25	RABT	Receive Abort Detect	0: Frame reception has not been aborted or no reception directive has been issued 1: Frame reception has been aborted	R/W
b26	TABT	Transmit Abort Detect	0: Frame transmission has not been aborted or no transmission directive has been issued 1: Frame transmission has been aborted	R/W
b29 to b27	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b30	TWB	Write-Back Complete	0: Write-back has not been completed, or no transmission directive has been issued 1: Write-back has been completed	R/W
b31	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. The ECI bit is read-only. When the source in ECSR in the ETHERC is cleared, the ECI bit is also cleared.
Note 2. When an address error is detected, the EDMAC halts transmission or reception. To resume the operation, execute a software reset with the EDMR.SWR bit.

EESR shows communications status on the EDMAC and the ETHERC.
The information in EESR is reported in the form of interrupt sources. Individual bits are cleared by writing 1 (however, the ECI bit is a read-only bit that is not cleared by writing 1) and are not affected by writing 0. Each interrupt source can be masked by means of the corresponding bit in the ETHERC/EDMAC status interrupt permission register (EESIPR).

RTSF Bit (Receive Too-Short Frame)

This bit indicates that a frame of fewer than 64 bytes has been received.

RTLF Bit (Receive Too-Long Frame)

This bit indicates that a frame whose byte size exceeds the upper limit for the receive frame length set by the receive frame length register (RFLR) in the ETHERC has been received.

TRO Bit (Transmit Retry Over)

This bit indicates that a retry-over condition has occurred during frame transmission. A total of 16 transmission retries including 15 retries based on the back-off algorithm have failed after the ETHERC transmission starts.

CD Bit (Delayed Collision Detect)

This bit indicates that a delayed collision has been detected during frame transmission.

DLC Bit (Detect Loss of Carrier)

This bit indicates that loss of the carrier has been detected during frame transmission.

RFOF Bit (Receive FIFO Overflow)

This bit indicates that the receive FIFO has overflowed during frame reception.

RDE Bit (Receive Descriptor Empty)

When receive descriptor empty (RDE = 1) occurs, reception can be resumed by setting the RD0.RACT bit of the receive descriptor to 1 and then restarting the receive operation.

FR Bit (Frame Reception)

This bit indicates that a frame has been received and the receive descriptor has been updated. The FR bit is set to 1 each time a frame is received.

TFUF Bit (Transmit FIFO Underflow)

This bit indicates that the transmit FIFO has underflowed during frame transmission. Incomplete data is sent onto the line.

TDE Bit (Transmit Descriptor Empty)

This bit indicates that the transmit descriptor active bit (TD0.TACT) of a transmit descriptor read by the EDMAC is not set if the previous descriptor does not represent the end of a frame in multi-buffer frame processing. As a result, an incomplete frame may be sent.
When transmit descriptor empty (TDE = 1) occurs, execute a software reset and initiate transmission. In this case, transmission starts from the address that is stored in TDLAR.

TC Bit (Frame Transmit Complete)

This bit indicates that all the data specified by the transmit descriptor has been transmitted from the ETHERC. The TC bit is set to 1, assuming the completion of transmission, when transmission of one frame is completed in single-frame/ single-buffer operation or when the last data of a frame has been transmitted and the transmit descriptor active bit (TD0.TACT) of the next descriptor is not set in multi-buffer frame processing. After frame transmission, the EDMAC writes the transmission status back to the relevant descriptor.

ADE Bit (Address Error)

This bit indicates that the memory address that the EDMAC tried to transfer is found illegal.
RFCOF Bit (Receive Frame Counter Overflow)
This bit indicates that the frame counter in the receive FIFO has overflowed.

RABT Bit (Receive Abort Detect)

This bit indicates that the ETHERC has aborted receiving a frame because of failures during frame reception.

TABT Bit (Transmit Abort Detect)

This bit indicates that the ETHERC has aborted transmitting a frame because of failures during frame transmission.

TWB Bit (Write-Back Complete)

This bit indicates that write-back from the EDMAC to the corresponding descriptor after frame transmission has completed. This operation is enabled only when the transmit interrupt setting (TIS) bit in the transmit interrupt setting register (TRIMD) is set to 1.

32.2.7 ETHERC/EDMAC Status Interrupt Permission Register (EESIPR)

Address(es): 000C 0030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP	ADEIP	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 on Received Frame Interrupt Enable	0: CRC error interrupt is disabled 1: CRC error interrupt is enabled	R/W
b1	PREIP	PHY-LSI Receive Error Interrupt Enable	0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled	R/W
b2	RTSFIP	Receive Too-Short Frame Interrupt Enable	0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled	R/W
b3	RTLFIP	Receive Too-Long Frame Interrupt Enable	0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled	R/W
b4	RRFIP	Receive Residual-Bit Frame Interrupt Enable	0: Receive residual-bit frame interrupt is disabled 1: Receive residual-bit frame interrupt is enabled	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	RMAFIP	Receive Multicast Address Frame Interrupt Enable	0: Receive multicast address frame interrupt is disabled 1: Receive multicast address frame interrupt is enabled	R/W
b8	TROIP	Transmit Retry Over Interrupt Enable	0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled	R/W
b9	CDIP	Delayed Collision Detect Interrupt Enable	0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled	R/W
b10	DLCIP	Detect Loss of Carrier Interrupt Enable	0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled	R/W
b11	CNDIP	Carrier Not Detect Interrupt Enable	0: Carrier not detect interrupt is disabled 1: Carrier not detect interrupt is enabled	R/W
b15 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b16	RFOFIP	Receive FIFO Overflow Interrupt Enable	0: Overflow interrupt is disabled 1: Overflow interrupt is enabled	R/W
b17	RDEIP	Receive Descriptor Empty Interrupt Enable	0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled	R/W
b18	FRIP	Frame Reception Interrupt Enable	0: Frame reception interrupt is disabled 1: Frame reception interrupt is enabled	R/W
b19	TFUFIP	Transmit FIFO Underflow Interrupt Enable	0: Underflow interrupt is disabled 1: Underflow interrupt is enable	R/W
b20	TDEIP	Transmit Descriptor Empty	0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled	R/W
b21	TCIP	Frame Transmission Complete Interrupt Enable	0: Frame transmission complete interrupt is disabled 1: Frame transmission complete interrupt is enabled	R/W
b22	ECIIP	ETHERC Status Register Source Interrupt Enable	0: ETHERC status interrupt is disabled 1: ETHERC status interrupt is enabled	R/W
b23	ADEIP	Address Error Interrupt Enable	0: Address error interrupt is disabled 1: Address error interrupt is enabled	R/W

Bit	Symbol	Bit Name	Description	R/W
b24	RFCOFIP	Receive Frame Counter Overflow Interrupt Enable	0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled	R/W
b25	RABTIP	Receive Abort Detect Interrupt Enable	0: Receive abort detect interrupt is disabled 1: Receive abort detect interrupt is enabled	R/W
b26	TABTIP	Transmit Abort Detect Interrupt Enable	0: Transmit abort detect interrupt is disabled 1: Transmit abort detect interrupt is enabled	R/W
b29 to b27	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b30	TWBIP	Write-Back Complete Interrupt Enable	0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled	R/W
b31	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

EESIPR enables interrupts corresponding to individual bits in the ETHERC/EDMAC status register (EESR). An interrupt is enabled by writing 1 to the corresponding bit.

32.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)

Address(es): 000C 0038h

	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
	—	—	—	—	CNDCE	DLCCE	CDCE	TROCE	RMAFCE	—	—	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE
Value after 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	CERFCE	CERF Bit Copy Directive	0: Reflects the EESR.CERF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b1	PRECE	PRE Bit Copy Directive	0: Reflects the EESR.PRE bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b2	RTSFCE	RTSF Bit Copy Directive	0: Reflects the EESR.RTSF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b3	RTLFCE	RTLF Bit Copy Directive	0: Reflects the EESR.RTLF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b4	RRFCE	RRF Bit Copy Directive	0: Reflects the EESR.RRF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	RMAFCE	RMAF Bit Copy Directive	0: Reflects the EESR.RMAF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b8	TROCE	TRO Bit Copy Directive	0: Reflects the EESR.TRO bit status in the TD0.TFS bit in the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TD0.TFS bit in the transmit descriptor	R/W
b9	CDCE	CD Bit Copy Directive	0: Reflects the EESR.CD bit status in the TD0.TFS bit in the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TD0.TFS bit in the transmit descriptor	R/W
b10	DLCCE	DLC Bit Copy Directive	0: Reflects the EESR.DLC bit status in the TD0.TFS bit in the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TD0.TFS bit in the transmit descriptor	R/W
b11	CNDCE	CND Bit Copy Directive	0: Reflects the EESR.CND bit status in the TD0.TFS bit in the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TD0.TFS bit in the transmit descriptor	R/W
b31 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TRSCER specifies whether the information for the transmit and receive state reported by bits in the ETHERC/EDMAC status register (EESR) is to be reflected in the TFS25 to TFS0 or RFS26 to RFS0 bits of the corresponding descriptor. The bits in TRSCER correspond to bits 11 to 0 in EESR. When a bit is cleared to 0, the transmit status (bits 11 to 8 in EESR) is reflected in the TFS3 to TFS0 bits of the transmit descriptor, and the receive status (bits 7 to 0 in EESR) is reflected in the RFS7 to RFS0 bits of the receive descriptor. When a bit is set to 1, the occurrence of the corresponding source is not reflected in the descriptor. After this LSI is reset, all bits are cleared to 0.

32.2.9 Receive Missed-Frame Counter Register (RMFCR)

Address(es): 000C 0040h

	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	These bits are always read as 0. The write value should always be 0.	R/W

RMFCR indicates the number of frames that could not be stored in the receive buffer and so were discarded during reception. When the receive FIFO overflows, the receive frames in the FIFO are discarded. The number of frames discarded at this time is counted. When the value in RMFCR reaches FFFFh, count-up is halted. The counter value is cleared to 0 by a write to RMFCR with any value.

32.2.10 Transmit FIFO Threshold Register (TFTR)

Address(es): 000C 0048h

	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: 52 bytes 00Eh: 56 bytes : : 01Fh: 124 bytes 020h: 128 bytes : : 03Fh: 252 bytes 040h: 256 bytes : : 07Fh: 508 bytes 080h: 512 bytes : : 0FFh: 1020 bytes 100h: 1024 bytes : : 1FFh: 2044 bytes 200h: 2048 bytes 201h to 7FFh: Setting prohibited	R/W
b31 to b11	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. When starting transmission before one frame of data write has completed, take care no underflow occurs.
Note 2. Operation cannot be guaranteed when the value set in TFTR is greater than the transmit FIFO size.
Note 3. To prevent a transmit underflow, setting the initial value (store and forward modes) is recommended.

TFTR specifies the transmit FIFO threshold at which the first transmission is started. The actual threshold is four times the set value.
The ETHERC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes specified in TFTR, when the transmit FIFO is full, or when one frame of data write is performed. TFTR must be set in the transmission-halt state.

TFT[10:0] Bits (Transmit FIFO Threshold)

The transmit FIFO threshold must be set to a value smaller than the FIFO size specified by the FIFO depth register (FDR).

32.2.11 FIFO Depth Register (FDR)

Address(es): 000C 0050h

	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 Size	00000: 256 bytes 00001: 512 bytes 00010: 768 bytes 00011: 1024 bytes 00100: 1280 bytes 00101: 1536 bytes 00110: 1792 bytes 00111: 2048 bytes Other than the above: Setting prohibited	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b12 to b8	TFD[4:0]	Transmit FIFO Size	00000: 256 bytes 00001: 512 bytes 00010: 768 bytes 00011: 1024 bytes 00100: 1280 bytes 00101: 1536 bytes 00110: 1792 bytes 00111: 2048 bytes Other than the above: Setting prohibited	R/W
b31 to b13	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. Operation cannot be guaranteed when the value set in FDR is greater than the transmit or receive FIFO size.

FDR specifies the sizes of the FIFO buffers for transmission and reception.
When activating the ETHERC module, set this register to 0000 0707h.

RFD[4:0] Bits (Receive FIFO Size)

These bits specify the size of the receive FIFO. The setting must not be changed after transmission/reception has started.

TFD[4:0] Bits (Transmit FIFO Size)

These bits specify the size of the transmit FIFO. The setting must not be changed after transmission/reception has started.

32.2.12 Receiving Method Control Register (RMCR)

Address(es): 000C 0058h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNC	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 Bit Reset	0: Allows the hardware to reset the receive request bit (RR) in EDRRR automatically upon completion of reception of one frame. Control is possible for each frame. To receive the subsequent receive frame, the RR bit needs to be set again. 1: Allows the higher-level software to control the receive request bit (RR) in EDRRR. Once the RR bit is set to 1, the hardware continues to fetch the receive descriptor and receive frames automatically until the RR bit is cleared to 0. In other words, continuous reception of multiple frames is possible. It is recommended to set the RNR bit to 1 when continuous reception is used. However, when a receive descriptor empty is detected, the hardware clears the RR bit automatically.	R/W
b1	RNC	Receive Request Bit Non-Reset Mode	0: Nop 1: Allows the software to reset the receive request bit (RR) in EDRRR. In this case, even when the RD0.RACT bit in the fetched descriptor is 0 (receive descriptor empty), the RR bit is not automatically reset and the receive descriptor is continuously fetched to continue DMA transfers of the receive frames.	R/W
b31 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RMCR specifies the control method for the receive request (RR) bit in the EDMAC receive request register (EDRRR) while a frame is received.
RMCR must be set in the receiving-halted state.

32.2.13 Transmit FIFO Underrun Counter (TFUCR)

Address(es): 000C 0064h

	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	Indicates the count of underflows having occurred in the transmit FIFO. The counter stops when the count value reaches FFFFh.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TFUCR indicates the count of underruns having occurred in the transmit FIFO.
The count value is cleared to 0 by writing any value to this register.

32.2.14 Receive FIFO Overflow Counter (RFOCR)

Address(es): 000C 0068h

	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	Indicates the count of overflows having occurred in the receive FIFO. The counter stops when the count value reaches FFFFh.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RFOCR indicates the count of overflows having occurred in the receive FIFO.
The count value is cleared to 0 by writing any value to this register.

32.2.15 Independent Output Signal Setting Register (IOSR)

Address(es): 000C 006Ch

	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 ET_EXOUT pin outputs a low level. 1: The ET_EXOUT pin outputs a high level.	R/W
b31 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

IOSR selects the output level of the external output pin (ET_EXOUT) in external loopback mode. The ET_EXOUT pin can be used to specify 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 ET_EXOUT pin.

32.2.16 Flow Control Start FIFO Threshold Setting Register (FCFTR)

Address(es): 000C 0070h

	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 Overflow BSY Output Threshold	b2 b0 0 0 0: When 256 – 32 bytes of data is stored in the receive FIFO. 0 0 1: When 512 – 32 bytes of data is stored in the receive FIFO. : 1 1 0: When 1792 – 32 bytes of data is stored in the receive FIFO. 1 1 1: When 2048 – 64 bytes of data is stored in the receive FIFO.	R/W
b15 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b18 to b16	RFFO[2:0]	Receive Frame Count Overflow BSY Output Threshold	b18 b16 0 0 0: When two receive frames have been stored in the receive FIFO. 0 0 1: When four receive frames have been stored in the receive FIFO. 0 1 0: When six 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	These bits are always read as 0. The write value should always be 0.	R/W

FCFTR specifies the flow control of the ETHERC (specifies the threshold of automatic PAUSE output).
The threshold can be set in terms of the data size in the receive FIFO (RFDO[2:0] bits) and the number of receive frames (RFFO[2:0] bits). Flow control is turned on when either of the data size in the receive FIFO or the number of receive frames satisfies the corresponding threshold condition.
If the same receive FIFO size as set by the FIFO depth register (FDR) is set in FCFTR when flow control is to be turned on according to the condition set in the RFDO[2:0] bits, flow control is turned on with (FIFO data size – 64) bytes. For instance, when the FDR.RFD[4:0] bits = 00111b and the FCFTR.RFDO[2:0] bits = 111, flow control is turned on when (2,048 – 64) bytes of data is stored in the receive FIFO. The value set in the RFDO[2:0] bits should be equal to or smaller than the value set in the FDR.RFD[4:0] bits.

32.2.17 Receive Data Padding Insert Register (RPADIR)

Address(es): 000C 0078h

	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: Inserts the specified padding size immediately before the first byte of the receive data. 01h: Inserts the specified padding size immediately before the second byte of the receive data. : 3Eh: Inserts the specified padding size immediately before the 63rd byte of the receive data. 3Fh: Inserts the specified padding size immediately before the 64th byte of the receive data.	R/W
b15 to b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b17, b16	PADS[1:0]	Padding Size	b17 b16 0 0: No padding insertion 0 1: 1-byte insertion 1 0: 2-byte insertion 1 1: 3-byte insertion	R/W
b31 to b18	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RPADIR specifies padding insertion in receive data.

Before modifying the settings of RPADIR, execute a software reset through the EDMR.SWR bit.

32.2.18 Transmit Interrupt Setting Register (TRIMD)

Address(es): 000C 007Ch

	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 Setting	0: Interrupt not set No interrupt notification is sent in the mode selected by the TIM bit. When the TIS bit is 0, the TIM bit setting is invalid. 1: Interrupt set An interrupt notification is sent by setting the EESR.TWB bit to 1 in the mode selected by the TIM bit.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	TIM	Transmit Interrupt Mode	0: Per-transmit-frame mode An interrupt is issued upon write-back completion of each frame. 1: Interrupt mode An interrupt is issued upon write-back completion of the transmit descriptor with the TD0.TWBI bit set to 1.	R/W
b31 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TRIMD specifies whether to notify write-back completion of each frame during transmission by means of the EESR.TWB bit and an interrupt.

32.2.19 Receive Buffer Write Address Register (RBWAR)

Address(es): 000C 00C8h

	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

RBWAR stores the address of data to be written in the receive buffer by the EDMAC.
Which addresses in the receive buffer are processed by the EDMAC can be recognized by monitoring addresses indicated in RBWAR. The address to which the EDMAC is actually writing may be different from the value read from RBWAR.
RBWAR is read-only. Writing to this register is prohibited.

32.2.20 Receive Descriptor Fetch Address Register (RDFAR)

Address(es): 000C 00CCh

	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

RDFAR stores the descriptor start address that is required when the EDMAC fetches descriptor information from the receive descriptor.
Which receive descriptor information is used for processing by the EDMAC can be recognized by monitoring addresses indicated in RDFAR. The address from which the EDMAC is actually fetching a descriptor may be different from the value read from RDFAR.
RDFAR is read-only. Writing to this register is prohibited.

32.2.21 Transmit Buffer Read Address Register (TBRAR)

Address(es): 000C 00D4h

	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

TBRAR stores the address of data to be read from the transmit buffer by the EDMAC.
Which addresses in the transmit buffer are processed by the EDMAC can be recognized by monitoring addresses indicated in TBRAR. The address from which the EDMAC is actually reading may be different from the value read from TBRAR.
TBRAR is read-only. Writing to this register is prohibited.

32.2.22 Transmit Descriptor Fetch Address Register (TDFAR)

Address(es): 000C 00D8h

	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

TDFAR stores the descriptor start address that is required when the EDMAC fetches descriptor information from the transmit descriptor.
Which transmit descriptor information is used for processing by the EDMAC can be recognized by monitoring addresses indicated in TDFAR. The address from which the EDMAC is actually fetching a descriptor may be different from the value read from TDFAR.
TDFAR is read-only. Writing to this register is prohibited.

32.3 Operation

The EDMAC, connected to the ETHERC, allows efficient transfer of transmit/receive data between the ETHERC and memory (buffers) without CPU intervention. The EDMAC reads the control information, called descriptors, by itself. The descriptors corresponding to each buffer hold buffer pointers and other information. The EDMAC reads transmit data from the transmit buffer and writes receive data to the receive buffer according to the control information. By arranging such multiple descriptors continuously (i.e., making a descriptor list), data can be transmitted or received sequentially.

32.3.1 Descriptor Lists and Data Buffers

By the communication program, a transmit descriptor list and a receive descriptor list should be created in memory space prior to transmission and reception. The start addresses of these lists should be set in the transmit descriptor list start address register and receive descriptor list start address register.
The start addresses of the descriptor lists should be placed on the address boundaries in accordance with the descriptor length specified in the EDMAC mode register (EDMR). Here, the start address of the transmit buffer can be placed on a longword, word, or byte boundary.

32.3.1.1 Transmit Descriptor

Figure 32.2 shows the relationship between a transmit descriptor and a transmit buffer. The transmit descriptor can relate one transmit frame to one transmit buffer (single-frame/single-buffer operation) or one frame to multiple transmit buffers (single-frame/multi-buffer operation).
When the transmit buffer length (TBL) is to be set to 1 to 16 bytes, the buffer address needs to be placed on a 32-byte boundary. When the transmit buffer length (TBL) is set to 0 byte, operation cannot be guaranteed.
Each transmit descriptor is cleared to 0000 0000h by a reset.

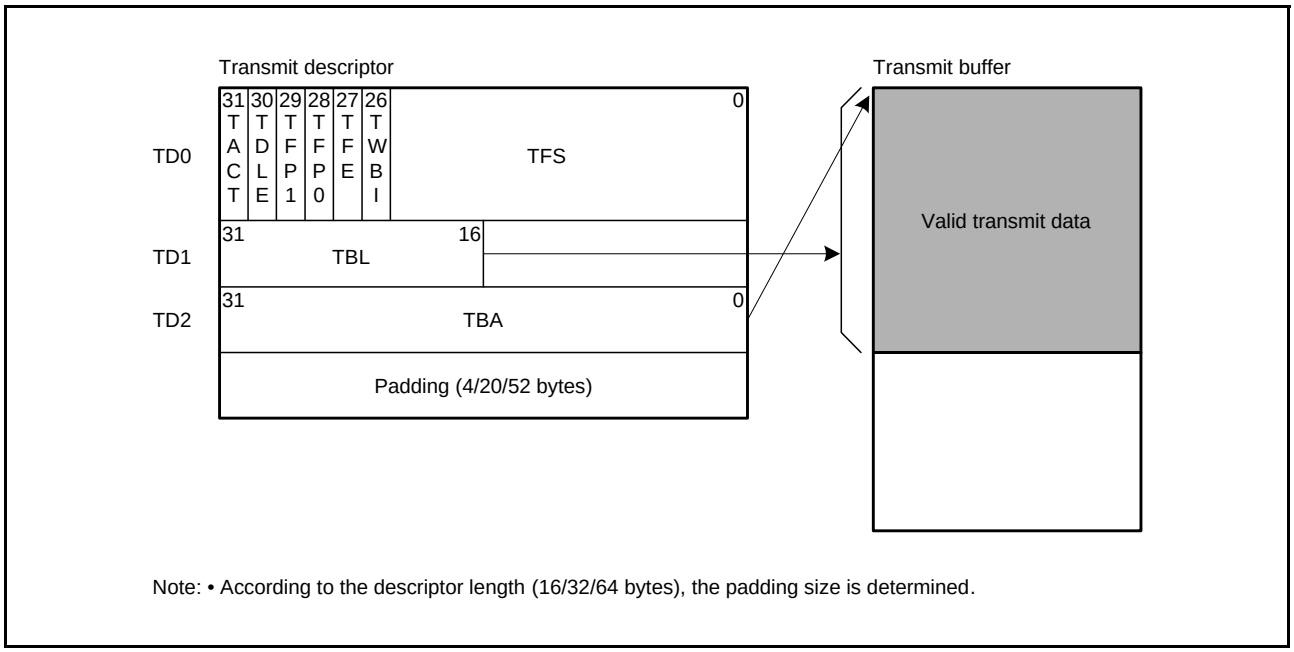


Figure 32.2 Relationship between Transmit Descriptor and Transmit Buffer

(1) Transmit Descriptor 0 (TD0)

Bit	Symbol	Bit Name	Description	R/W
b25 to b0	TFS	Transmit Frame Status	TFS25 toTFS9: Reserved (In writing, the value should always be 0.) TFS8: Detect Transmit Abort When set to 1, this bit indicates that the abort signal is set to 1 during frame transmission (causing TFE to be set). TFS7 to TFS4: Reserved (In writing, the value should always be 0.) TFS3: Detect No Carrier (corresponding to the EESR.CND bit) TFS2: Detect Loss of Carrier (corresponding to the EESR.DLC bit) TFS1: Detect Delayed Collision during Transmission (corresponding to the EESR.CD bit) TFS0: Transmit Retry Over (corresponding to the EESR.TRO bit) When set to 1, these bits indicate that TFS8 to TFS1 have been set to 1 during frame transmission. (Although TFE is normally set when these bits are set to 1, it can be prevented from being set by so setting TRSCER.)	R/W
b26	TWBI	Write-Back Completion Interrupt Notification	(This bit is valid when TRIMD is set so.) 0: Nop 1: An interrupt is generated upon completion of write-back to this descriptor.	R/W
b27	TFE	Transmit Frame Error	0: Frame transmission is continued (normal operation). 1: Frame transmission has been aborted.	R/W
b29, b28	TFP[1:0]	Transmit Frame Position	b29 b28 0 0: Transmission of the frame of the transmit buffer specified by this descriptor is continued. (The frame is incomplete.) 0 1: The transmit buffer specified by this descriptor contains the end of the frame (The frame is complete.) 1 0: The transmit buffer specified by this descriptor is the start of the frame (The frame is incomplete.) 1 1: The contents in the transmit buffer specified by this descriptor correspond to one frame (single-frame/single-buffer).	R/W
b30	TDLE	Transmit Descriptor Ring End	When set to 1, the TDLE bit indicates that the corresponding descriptor is the last one of the descriptor ring.	R/W
b31	TACT	Transmit Descriptor Active	Indicates that the corresponding descriptor is active.	R/W

Note: • The underlined bits are subject to write-back.

TD0 indicates the transmit frame status, informing frame transmission status.

TFE Bit (Transmit Frame Error)

When set to 1, the TFE bit indicates that an error is indicated by any of the TFS bits. (Through the TRSCER setting, it is possible to prevent this bit from being set by an event indicated by TFS7 to TFS0. It is impossible, however, if an event indicated by TFS7 to TFS0 also causes TFS8 to be set.)

TFP[1:0] Bits (Transmit Frame Position)

These bits relate the transmit buffer to the transmit frame. The settings of the TFP bits and the TBL bits should be logically correct in the consecutive descriptors.

TACT Bit (Transmit Descriptor Active)

This bit indicates that the corresponding descriptor is active. The TACT bit is set to 1 by software. This bit is cleared to 0 by hardware when a transmit frame has been completely transferred or when transmission has been aborted due to some cause.

(2) Transmit Descriptor 1 (TD1)

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b31 to b16	TBL	Transmit Buffer Length	Indicates the length of the relevant transmit buffer in terms of valid bytes.	R/W

TD1 indicates the length of the relevant transmit buffer in terms of valid bytes.

(3) Transmit Descriptor 2 (TD2)

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TBA	Transmit Buffer Address	Indicates the start address of the transmit buffer.	R/W

TD2 indicates the start address of the transmit buffer.

32.3.1.2 Receive Descriptor

Figure 32.3 shows the relationship between a receive descriptor and a receive buffer. The receive buffer address should be placed on a 32-byte boundary.
When the receive buffer length (RBL) is set to 0 byte, operation specified by the descriptor cannot be guaranteed.
Each receive descriptor is cleared to 0000 0000h by a reset.

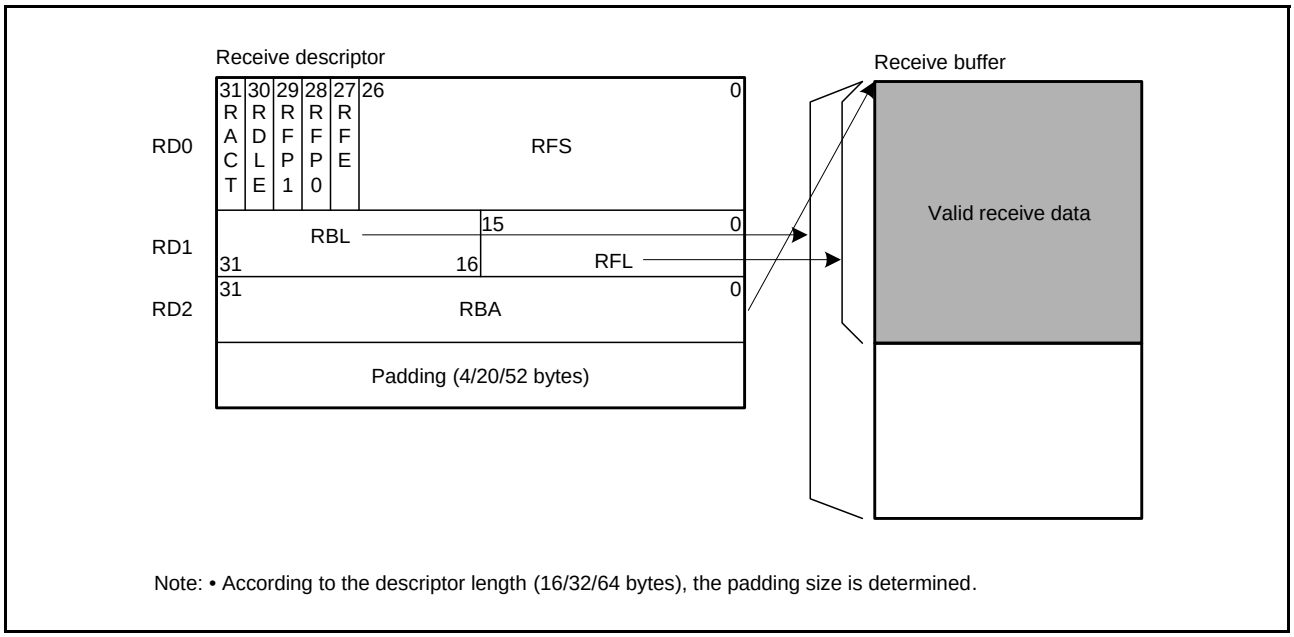


Figure 32.3 Relationship between Receive Descriptor and Receive Buffer

(1) Receive Descriptor 0 (RD0)

Bit	Symbol	Bit Name	Description	R/W
b26 to b0	RFS	Receive Frame Status	RF26 to RF10: Reserved (In writing, the value should always be 0.) RFS9: Receive FIFO Overflow (corresponding to the EESR.RFOF bit) When set to 1, this bit indicates that a receive FIFO overflow has occurred terminating the frame halfway and that the frame has been written back (causing RFE to be set). TFS8: Detect Receive Abort When set to 1, this bit indicates that the abort signal is set to 1 during frame transmission (causing RFE to be set). RFS7: Receive Multicast Address Frame (corresponding to the EESR.RMAF bit) RFS6 and RFS5: Reserved (In writing, the value should always be 0.) RFS4: Receive Residual-Bit Frame (corresponding to the EESR.RRF bit) RFS3: Receive Too-Long Frame (corresponding to the EESR.RTLF bit) RFS2: Receive Too-Short Frame (corresponding to the EESR.RTSF bit) RFS1: PHY-LSI Receive Error (corresponding to the EESR.PRE bit) RFS0: CRC Error on Received Frame (corresponding to the EESR.CERF bit) When set to 1, these bits indicate that RFS8 to RFS1 have been set to 1 during frame reception. (Although RFE is normally set when these bits are set to 1, it can be prevented from being set by so setting TRSCER.)	R/W
b27	RFE	Receive Frame Error	0: No error in RFS 1: An error has been indicated in RFS	R/W
b29, b28	RFP[1:0]	Receive Frame Position	b29 b28 0 0: Reception of the frame of the receive buffer specified by this descriptor is continued. (The frame is incomplete.) 0 1: The receive buffer specified by this descriptor contains the end of the frame (The frame is complete.) 1 0: The receive buffer specified by this descriptor is the start of the frame (The frame is incomplete.) 1 1: The contents in the receive buffer specified by this descriptor correspond to one frame (single-frame/single-buffer).	R/W
b30	RDLE	Receive Descriptor Ring End	When set to 1, the RDLE bit indicates that the corresponding descriptor is the last one of the descriptor ring.	R/W
b31	RACT	Receive Descriptor Active	Indicates that the corresponding descriptor is active.	R/W

Note: • The underlined bits are subject to write-back.

RD0 indicates the receive frame status, informing frame reception status.

RFE Bit (Receive Frame Error)

When set to 1, the RFE bit indicates that an error is indicated by any of the RFS bits. (Through the TRSCER setting, it is possible to prevent this bit from being set by an event indicated by RFS7 to RFS0. It is impossible, however, if an event indicated by RFS7 to RFS0 also causes RFS8 to be set.)

RFP[1:0] Bits (Receive Frame Position)

These bits relate the receive buffer to the receive frame.

RACT Bit (Receive Descriptor Active)

This bit indicates that the corresponding descriptor is active. The RACT bit is set to 1 by software. This bit is cleared to 0 by hardware when an entire receive frame has been completely transferred to the buffer address specified 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 Data Length	Indicates the length of (number of bytes in) a receive frame stored in the buffer. The number of bytes for padding insertion specified by RPADIR is excluded. These bits are written back to the descriptor containing the end of a frame.	R/W
b31 to b16	RBL	Receive Buffer Length	Indicates the length of the relevant receive buffer in terms of bytes. The buffer length should be set to n × 32.	R/W

Note: • The underlined bits are subject to write-back.

RD1 indicates the receive buffer length and receive frame length.

(3) Receive Descriptor 2 (RD2)

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBA	Receive Buffer Address	Indicates the start address of the receive buffer. The buffer address should be set on a 32-byte boundary.	R/W

RD2 indicates the start address of the receive buffer.

32.3.2 Transmission

When the EDTRR.TR bit is set to 1 while the transmission function is enabled, the EDMAC reads the descriptor following the previously used descriptor from the transmit descriptor list (or the descriptor indicated by the transmit descriptor start address register (TDLAR) at the initial start time). If the TD0.TACT bit of the read descriptor is set to 1 (active), the EDMAC sequentially reads transmit frame data from the transmit buffer start address specified by TD2 for transfer to the ETHERC. The ETHERC creates a transmit frame and starts transmission to the MII. After DMA transfer of data equivalent to the buffer length specified in the descriptor, the following processing is carried out according to the TD0.TFP value.
TD0.TFP[1:0] = 00 or 10 (frame continuation)
Descriptor write-back (writing to the TD0.TACT bit) is performed after DMA transfer.
TD0.TFP[1:0] = 01 or 11 (frame end)
Descriptor write-back (writing to the TD0.TACT bit and status bits) is performed after completion of frame transmission.

As long as the TD0.TACT bit of a read descriptor is set to 1 (active), the reading of EDMAC descriptors and the transmission of frames continue. When a descriptor with the TD0.TACT bit cleared to 0 (inactive) is read, the EDMAC clears the EDTRR.TR bit to 0 and completes transmit processing.

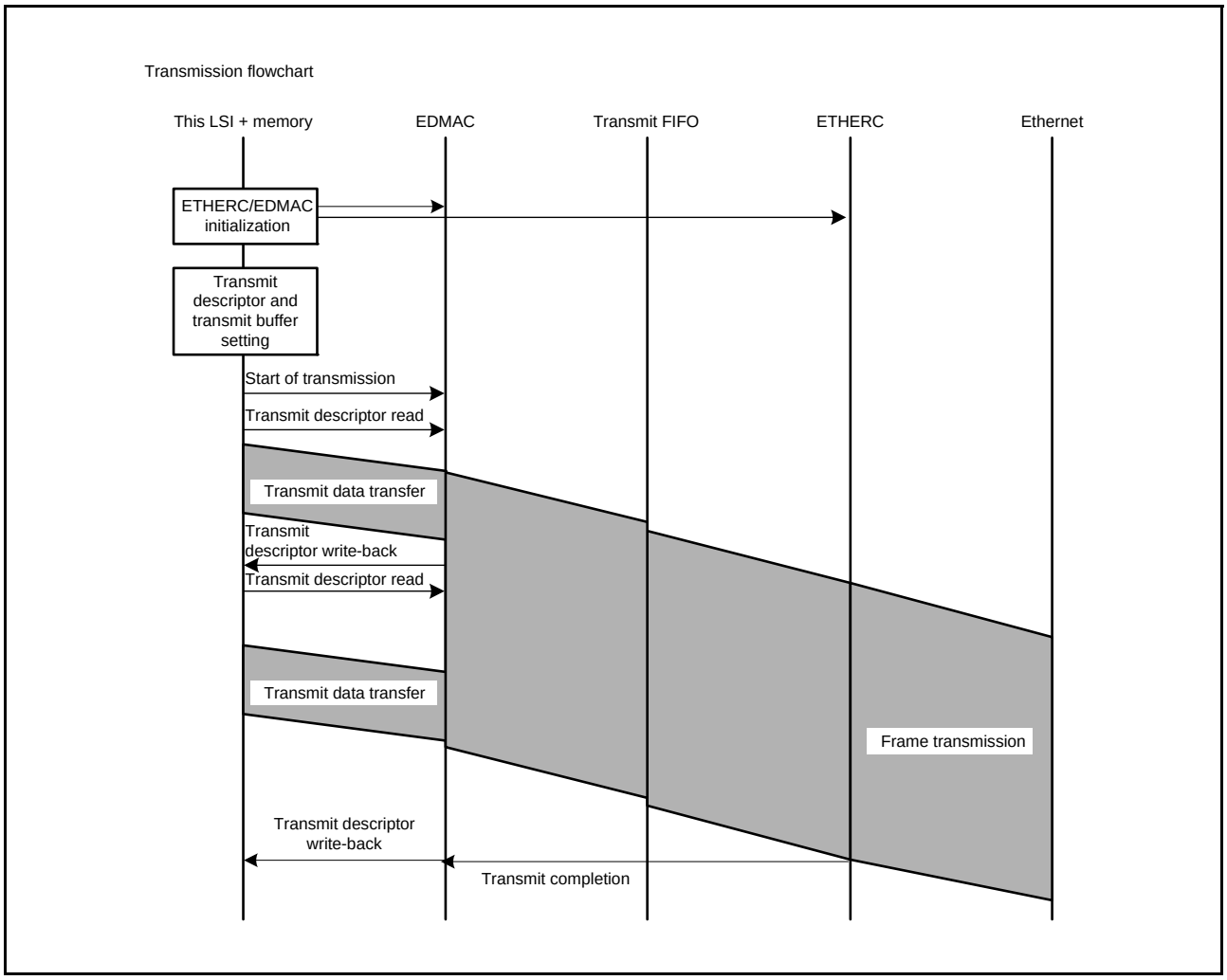


Figure 32.4 Sample Transmission Flowchart

32.3.3 Reception

When the CPU sets the EDRRR.RR bit while the receive function is enabled, the EDMAC reads the descriptor following the previously used descriptor from the receive descriptor list (or the descriptor indicated by the receive descriptor start address register (RDLAR) at the initial start time) then enters the receive standby state. Upon receiving the frame for this LSI while the RD0.RACT bit is set to 1 (active), the EDMAC transfers the frame to the receive buffer specified by RD2. If the data length of a received frame is longer than the buffer length specified by RD1, the EDMAC performs a write-back operation to the descriptor (with RD0.RFP[1:0] set to 10b or 00b) when the buffer becomes full, then reads the next descriptor. The EDMAC then continues to transfer data to the receive buffer specified by the new RD2. When frame reception is completed, or if frame reception is aborted because of a certain kind of error, the EDMAC performs write-back to the relevant descriptor (with RD0.RFP[1:0] set to 11b or 01b), and then ends the receive processing. The EDMAC then reads the next descriptor and enters the receive standby state again.
To receive frames continuously, the RNC bit must be set to 1 in the receive method control register (RMCR). The initial value is 0.

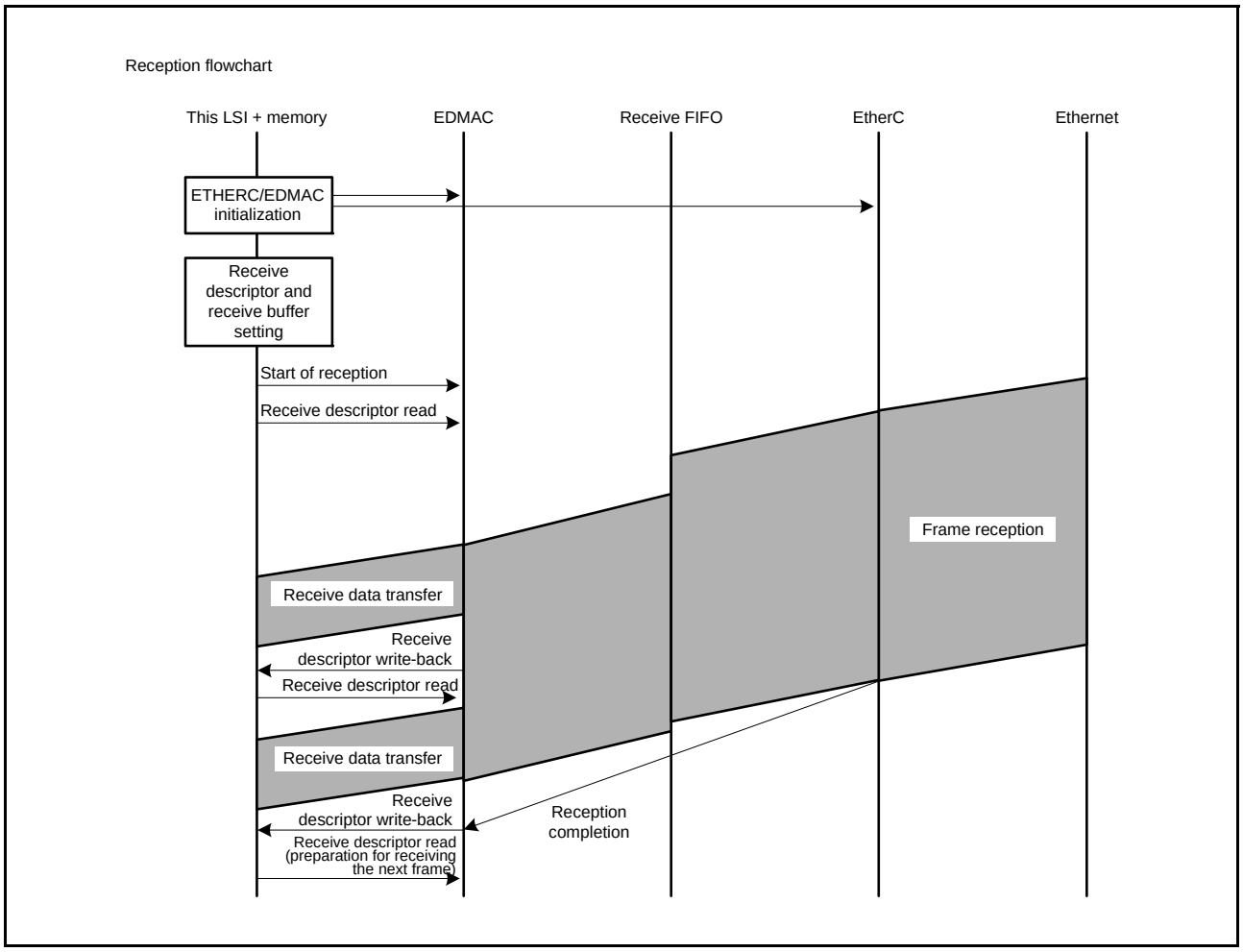


Figure 32.5 Sample Reception Flowchart

32.3.4 Transmit/Receive Processing of Multi-Buffer Frame

32.3.4.1 Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the processing shown in Figure 32.6 is carried out by the EDMAC.

In the figure, for the inactive transmit descriptors (TD0.TACT bit = 0), buffer data has already been transmitted successfully, and for the active transmit descriptors (TD0.TACT bit = 1), buffer data has not been transmitted. If a frame transmit error occurs in the first active descriptor (TD0.TACT bit = 1), transmission is halted immediately and the TD0.TACT bit is cleared to 0. The next descriptor is then read, and the position within the transmit frame is determined on the basis of the TD0.TFP[1:0] bits (continuing [00b] or end [01b]). In the case of a continuing descriptor, only the TD0.TACT bit is cleared to 0 and the next descriptor is read immediately. If the descriptor is the end descriptor, not only is the TD0.TACT bit cleared to 0, but write-back is also performed to the TD0.TFE and TD0.TFS bits at the same time. Data in the buffer is not transmitted between the occurrence of an error and write-back to the end descriptor. If error interrupts are enabled in the ETHERC/EDMAC status interrupt permission register (EESIPR), an interrupt is generated immediately after the write-back to the end descriptor.

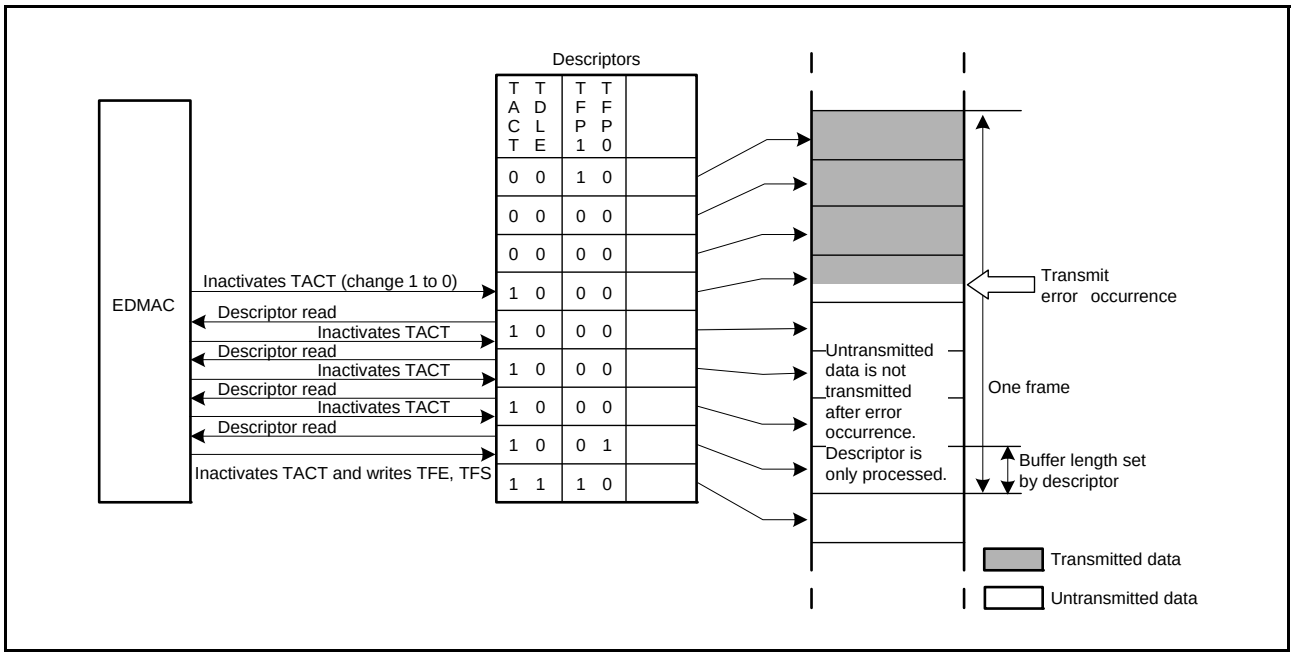


Figure 32.6 EDMAC Operation after Transmit Error

32.3.4.2 Receive Processing in Case of Multi-Buffer Frame

If an error occurs during multi-buffer frame reception, the EDMAC performs the processing shown in Figure 32.7. In the figure, for the inactive receive descriptors (RD0.RACT bit = 0), the buffers have received data successfully, and for the active receive descriptors (RD0.RACT bit = 1), the buffers have not received data. If a frame receive error occurs in the first active descriptor (RD0.RACT bit = 1) in the figure, the status is written back to the descriptor.
If error interrupts are enabled in EESIPR, an interrupt is generated immediately after the write-back. If there is a new frame receive request, reception is continued from the next buffer after the one in which the error occurred.

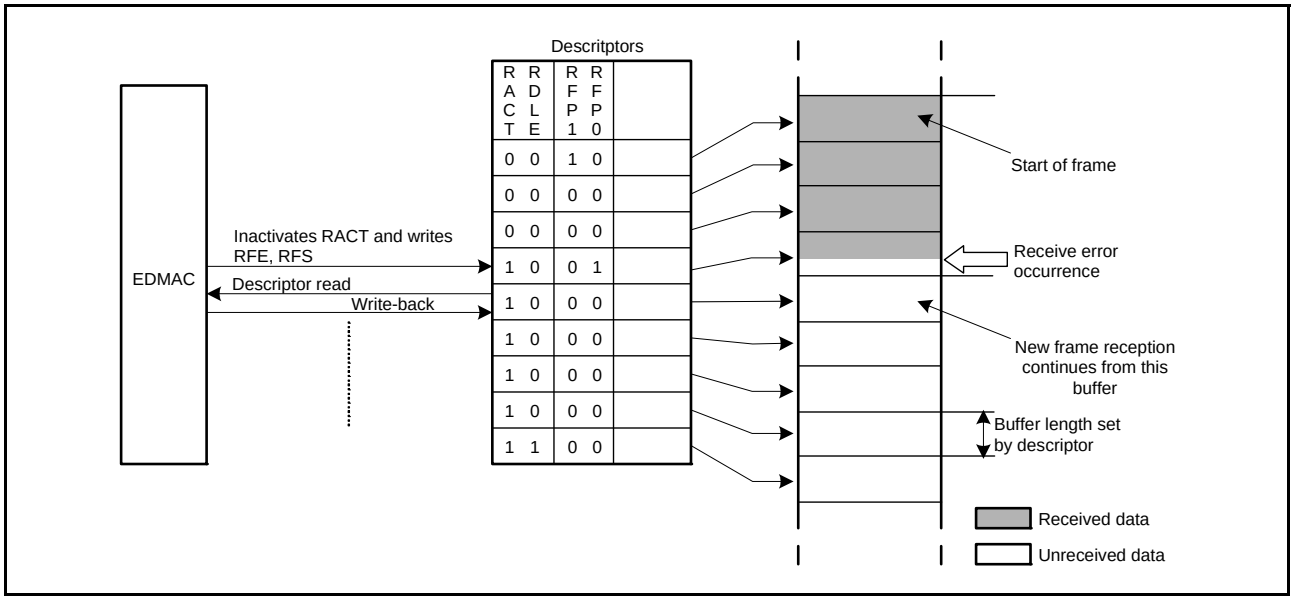


Figure 32.7 EDMAC Operation after Receive Error

33. USB 2.0 Host/Function Module (USBa)

33.1 Overview

Products of the RX63N/RX631 Group incorporate a USB2.0 host/function module (USB0) and a USB2.0 function module (USB1), for two USB ports.

The USB module is a USB controller that is equipped to operate as a host controller or function controller. As host controller or function controller, the module supports full-speed transfer as defined in revision 2.0 of the Universal Serial Bus Specification. The module has an internal USB transceiver and supports all of the transfer types defined in the USB specification.

The USB has buffer memory for data transfer, providing a maximum of ten pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE9, based on the peripheral devices or user system for communication.

Table 33.1 shows the specifications of the USB.

Table 33.1 Specifications of USB

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB2.0 are incorporated. • Two ports are provided. USB0 operates as a host controller or function controller, or in the OTG roles. USB1 operates as a function controller. • Software can switch between the host controller and function controller (can be switched by software). • Self-power mode or bus-power mode can be selected. • OTG (On-The-Go) is supported.
	When host controller operation is selected: • Full-speed transfer (12 Mbps) is supported • Communications with multiple peripheral devices connected via a single HUB • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers
	When function controller operation is selected: • Full-speed transfer (12 Mbps) 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
Internal bus interface	• Connected to internal peripheral bus 3
Pipe configuration	• Buffer memory for USB communications is provided. • Up to ten pipes can be selected (including the default control pipe). • Endpoint numbers can be assigned flexibly to PIPE1 to PIPE9.
	Transfer conditions that can be set for each pipe: • PIPE0: Control transfer only (default control pipe: DPC) Buffer size: 8, 16, 32, or 64 bytes (single buffer) • PIPE1 and PIPE2: Bulk transfer or isochronous transfer Buffer size: 8, 16, 32, or 64 bytes for bulk transfer or 1 to 256 bytes for isochronous transfer (double buffer can be specified) • PIPE3 to PIPE5: Bulk transfer only Buffer size: 8, 16, 32, or 64 bytes (double buffer can be specified) • PIPE6 to PIPE9: Interrupt transfer only Buffer size: 1to 64 bytes (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 or 1) port has been read (DCLRM) • NAK setting function for response PID generated by end of transfer (SHTNAK)
Power consumption reducing function	Module stop state can be set.

Figure 33.1 shows a block diagram of the USB.

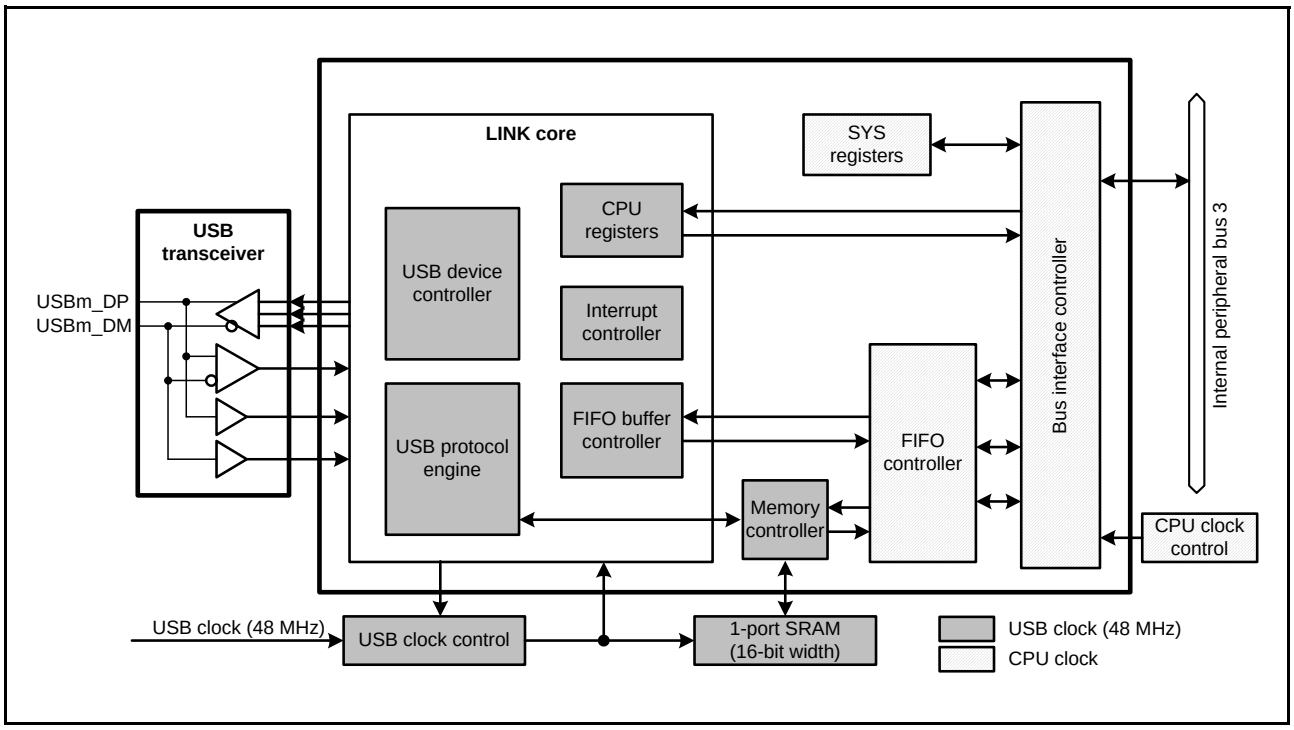


Figure 33.1 Block Diagram of USB

Table 33.2 shows the input/output pins of the USB.

Table 33.2 USB Pin Configuration

Port	Pin Name	I/O	Function
USB0	USB0_DP	I/O	D+ I/O pin of the port 0 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 port 0 USB on-chip transceiver This pin should be connected to the D– pin of the USB bus.
	USB0_VBUS	Input	Port 0 USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
	USB0_EXICEN	Output	Low-power control signal for port 0 external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for port 0 external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	Port 0 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	Mini-AB connector ID input signal should be connected to this pin when port 0 operates in OTG mode.
	USB0_DPUPE	Output	1.5-kΩ pull-up resistor control signal for USB D+ signal when port 0 operates as a function controller
	USB0_DPRPD USB0_DRPD	Output	15-kΩ pull-down resistor control signal for USB D+ and USB D– signals when port 0 operates as a host controller
USB1	USB1_DP	I/O	D+ I/O pin of the port 1 USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB1_DM	I/O	D– I/O pin of the port 1 USB on-chip transceiver This pin should be connected to the D– pin of the USB bus.
	USB1_VBUS	Input	Port 1 USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
	USB1_DPUPE	Output	1.5-kΩ pull-up resistor control signal for USB D+ signal when port 1 operates as a function controller
Common	VCC_USB	Input	USB power supply pin
	VSS_USB	Input	USB ground pin

33.2 Register Descriptions

33.2.1 System Configuration Control Register (SYSCFG)

Address(es): USB0.SYSCFG 000A 0000h, USB1.SYSCFG 000A 0200h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	—	—	DCFM*1	DRPD*1	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note 1. Bits b6 and b5 in the USB1.SYSCFG register are reserved.

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 always read as 0. The write value should always 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*1	D+/D- Line Resistor Control	0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.	R/W
b6	DCFM*1	Controller Function Select	0: Function controller function is selected. 1: Host controller function is selected.	R/W
b9 to b7	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b10	SCKE	USB Clock Enable	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 always read as 0. The write value should always be 0.	R/W

Note 1. Bits b6 and b5 in the USB1.SYSCFG register are reserved. These bits are read as 0. When writing, write 0 to these bits.

USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.

Modifying the USBE bit from 1 to 0 initializes some register bits as listed in Table 33.3.

This bit should be modified while the SCKE bit is 1.

When the host controller function is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] bits chattering, and checking that the USB bus state has been settled.

Table 33.3 Registers Initialized by Writing SYSCFG.USBE = 0

Selected Function	Register	Bit	Remarks
Function controller function	SYSSTS0	LNST[1:0]	The value is retained when the host controller function is selected.
	DVSTCTR0	RHST[2:0]	
	INTSTS0	DVSQ[2:0]	The value is retained when the host controller function is selected.
	USBADDR	USBADDR[6:0]	The value is retained when the host controller function is selected.
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	The value is retained when the host controller function is selected.
	USBVAL	WVALUE[15:0]	The value is retained when the host controller function is selected.
	USBINDX	WINDEX[15:0]	The value is retained when the host controller function is selected.
	USBLENG	WLENGTH[15:0]	The value is retained when the host controller function is selected.
Host controller function	DVSTCTR0	RHST[2:0]	
	FRMNUM	FRNM[10:0]	The value is retained when the function controller function is selected.

DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller function is selected.
When the DPRPU bit is set to 1 while the function controller function is selected, the USB asserts the USBm_DPUPE pin to notify the USB host of connection.
Setting the DPRPU bit to 1 when the function controller function is selected allows the USB to assert the USBm_DPUPE pin, thus notifying the USB host of connection. Modifying the DPRPU bit from 1 to 0 allows the USB to negate the USBm_DPUPE pin, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller function is selected, and should be set to 0 if the host controller function is selected.

DRPD Bit (D+/D- Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D- lines by the USB0_DPRPD and USB0_DRPD pins when the host controller function is selected.
This bit should be set to 1 if the host controller function is selected, and should be set to 0 if the function controller function is selected.

DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB.
This bit should be modified while both the DPRPU and DRPD bits are 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, DMA0PCFG, and DMA1PCFG can be read from and written to; the other registers in the USB cannot be read from or written to.

33.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): USB0.SYSSTS0 000A 0004h, USB1.SYSSTS0 000A 0204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OCVMON[1:0]*1		—	—	—	—	—	—	—	HTACT *1	—	—	—	IDMON *1	LNST[1:0]	
Value after reset:	0*2	0*2	0	0	0	0	0	0	0	0	0	0	0	0*2	0	0

Note 1. Bits b15, b14, b6, and b2 in the USB1.SYSSTS0 register are reserved.
Note 2. Depends on the USB0_OVRCURA/USB0_OVRCURB and USB0_ID pin status.

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor	b1 b0 0 0: SE0 0 1: J-State 1 0: K-State 1 1: SE1	R
b2	IDMON*1	External ID0 Input Pin Monitor	Indicates the status of the USB0_ID pin.	R
b5 to b3	—	Reserved	These bits are always read as 0 and cannot be modified.	R
b6	HTACT*1	USB Host Sequencer Status Monitor	0: Host sequencer of the USB0 is completely stopped. 1: Host sequencer of the USB0 is not completely stopped.	R
b13 to b7	—	Reserved	These bits are always read as 0 and cannot be modified.	R
b15, b14	OCVMON[1:0] *1	External USB0_OVRCURA/ USB0_OVRCURB Input Pin Monitor	The OCVMON[1] bit indicates the status of the USB0_OVRCURA pin. The OCVMON[0] bit indicates the status of the USB0_OVRCURB pin.	R

Note 1. Bits b15, b14, b6, and b2 in the USB1.SYSSTS0 register are reserved. These bits are read as 0. When writing, write 0 to these bits.

LNST[1:0] Bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the state of the USB data lines (D+ and D- lines).

The LNST[1:0] bits should be read after the connection processing (the SYSCFG.DPRPU bit = 1 is set) when the function controller function is selected; whereas after enabling pull-down of the lines (the SYSCFG.DRPD bit = 1 is set) when the host controller function is selected.

HTACT Bit (USB Host Sequencer Status Monitor)

The HTACT bit is 0 when the host sequencer of the USB0 is completely stopped.

Make sure the HTACT bit is 0 when stopping the clock supply to the USB0.

OCVMON[1:0] Bits (External USB0_OVRCURA/USB0_OVRCURB Input Pin Monitor)

The OCVMON[1:0] bits indicate the status of overcurrent from an external power-supply chip.

33.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): USB0.DVSTCTR0 000A 0008h, USB1.DVSTCTR0 000A 0208h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA *1	EXICEN *1	VBUSEN *1	WKUP	RWUPE *1	USBRST *1	RESUME *1	UACT*1	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note 1. Bits b11 to b9 and b7 to b4 in the USB1.DVSTCTR0 register are reserved.

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status	• When the host controller function 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*2 0 1 0: Full-speed connection x: Don't care • When the function controller function is selected b2 b0 0 0 0: Communication speed not determined 1 0 0: USB bus reset in progress 0 1 0: Full-speed connection	R
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b4	UACT*1	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 *1	Resume Output	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST*1	USB Bus Reset Output	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE*1	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 *3
b9	VBUSEN*1	USB0_VBUSEN Output Pin Control	The VBUSEN bit value is output as the status of the external USB0_VBUSEN pin without change.	R/W
b10	EXICEN*1	USB0_EXICEN Output Pin Control	The EXICEN bit value is output as the status of the external USB0_EXICEN pin without change.	R/W
b11	HNPBTOA*1	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 is set.	R/W
b15-b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. Bits b11 to b9 and b7 to b4 in the USB1.DVSTCTR0 register are reserved. These bits are read as 0. When writing, write 0 to these bits.
Note 2. The USB controller does not support communications with a low-speed device. When this value is read, abnormal connection processing should be executed in a higher application.
Note 3. Only 1 can be written.

RHST[2:0] Bits (USB Bus Reset Status)

The RHST[2:0] bits indicate the status of the USB bus reset.
When the host controller function is selected, the RHST[2:0] bits indicate 100b after software has written 1 to the USBRST bit.
The USB fixes the value of the RHST[2:0] bits when software writes 0 to the USBRST bit and the USB completes SE0 driving.
When the function controller function is selected, a DVST interrupt is generated as soon as the USB detects the USB bus reset and then the RHST[2:0] bits are fixed to 010b.

UACT Bit (USB Bus Enable)

The UACT bit enables operation of the USB0 bus (controls the SOF packet transmission to the USB bus) when the host controller function is selected.
With this bit set to 1, the USB0 puts the USB0 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 software has written 1 to the UACT bit.
With this bit set to 0, the USB0 enters the idle state after outputting SOF packets.
The USB0 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 function is selected.

RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller function is selected.
Setting the RESUME bit to 1 allows the USB0 to drive the port to the K-state and output the resume signal.
The USB0 continues outputting K-state while the RESUME bit = 1 (until software sets the RESUME bit to 0). The RESUME bit should be 1 (= resume period) for the time defined by the USB Specifications 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 function is selected.

USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller function is selected.
When the host controller function is selected, setting this bit to 1 allows the USB0 to drive SE0 of the USB port to reset the USB bus.
The USB0 continues outputting SE0 while USBRST = 1 (until software sets the USBRST bit to 0). The USBRST bit should be 1 (= USB bus reset period) for the time defined by the USB Specifications 2.0.
Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB0 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 function 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 function 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 ms) from the downstream port device and performs the resume processing (drives the port to the K-state).
With this bit set to 0, the USB0 ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.
While the PWUPE bit is 1, the internal clock should not be stopped even in the suspended state (the SYSCFG.SCKE bit should be set to 1).
This bit should be set to 0 if the function controller function 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 function is selected.
The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB clears this bit to 0 after outputting the 10-ms K-state.
According to the USB Specifications, 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 (bits INTSTS0.DVSQ[2:0] = 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 function 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 at FW to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

33.2.4 CFIFO Port Register (CFIFO)
D0FIFO Port Register (D0FIFO)
D1FIFO Port Register (D1FIFO)

Address(es): USB0.CFIFO 000A 0014h, USB0.D0FIFO 000A 0018h, USB0.D1FIFO 000A 001Ch
USB1.CFIFO 000A 0214h, USB1.D0FIFO 000A 0218h, USB1.D1FIFO 000A 021Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FIFOPORT[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	FIFOPORT[15:0]	FIFO Port	The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Table 33.4 and Table 33.5.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO ports. 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 features.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing the FIFO buffer using DMA transfer should be performed through the D0FIFO or D1FIFO port.
- The D1FIFO and D0FIFO ports can be accessed also by the CPU.
- When using functions specific to the 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 transfer function is used, etc.).
- Registers configuring a FIFO port do not affect other FIFO ports.
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

FIFOPORT[15:0] Bits (FIFO Port)

Accessing the FIFOPORT [15:0] bits allow 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 bit in each port control register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits in a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) as shown in Table 33.4 and Table 33.5.

Table 33.4 Endian Operation in 16-Bit Access

CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bit15 to 8	Bit7 to 0
0	N+1 data	N+0 data
1	N+0 data	N+1 data

Table 33.5 Endian Operation in 8-Bit Access

CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bit15 to 8	Bit7 to 0
0	Access prohibited*1	N+0 data
1	Access prohibited*1	N+0 data

Note 1. Reading from an access-prohibited area is not allowed.

33.2.5 CFIFO Port Select Register (CFIFOSEL)
D0FIFO Port Select Register (D0FIFOSEL)
D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): USB0.CFIFOSEL 000A 0020h, USB1.CFIFOSEL 000A 0220h

	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 Other than above: Setting prohibited	R/W
b4	—	Reserved	This bit is always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is always read as 0. The write value should always 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 always read as 0. The write value should always 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] bits are cleared when all of the receive data has been read from the CFIFO. (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] bits 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 cleared to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA 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 CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.
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 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 and MBW bits simultaneously.
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 bit 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): USB0.D0FIFOSEL 000A 0028h, USB0.D1FIFOSEL 000A 002Ch
USB1.D0FIFOSEL 000A 0228h, USB1.D1FIFOSEL 000A 022Ch

	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]	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 Other than above: Setting prohibited	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b11	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b12	DREQE	DMA Transfer Request Enable	0: DMA transfer request is disabled. 1: DMA 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] bits 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] bits 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 cleared to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA 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 CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

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 and MBW bits simultaneously.

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 Transfer Request Enable)

The DREQE bit enables or disables the DMA transfer request to be issued.

Before setting the DREQE bit to 1 to enable the DMA 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 out 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 bit 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.

33.2.6 CFIFO Port Control Register (CFIFOCTR)
D0FIFO Port Control Register (D0FIFOCTR)
D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USB0.CFIFOCTR 000A 0022h, USB0.D0FIFOCTR 000A 002Ah, USB0.D1FIFOCTR 000A 002Eh
USB1.CFIFOCTR 000A 0222h, USB1.D0FIFOCTR 000A 022Ah, USB1.D1FIFOCTR 000A 022Eh

	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	Indicates 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] bits shown below.	R
b12 to b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b13	FRDY	FIFO Port Ready	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Invalid 1: Clears the buffer memory on the CPU side.	R/W *1
b15	BVAL	Buffer Memory Valid Flag	0: Invalid 1: Writing ended	R/W *2

Note 1. Only 0 can be read.
Note 2. Only 1 can be written.

CFIFOCTR, D0FIFOCTR, and D1FIFOCTR are used for the corresponding FIFO ports.

DTLN[8:0] Bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.
While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT (n = 0, 1) bit value as described below.

- RCNT = 0
  The USB sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU (DMAC) 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] bits 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 Bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU (DMAC).
In the following cases, the USB sets the FRDY bit 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 NAK before setting the BCLR bit to 1.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL flag 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 bit in the FIFO port control register is 1 (set by the USB).

BVAL Flag (Buffer Memory Valid Flag)

The BVAL flag 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 flag 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 flag to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL flag 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 flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL flag should be done while the FRDY bit is 1 (set by the USB).
When the selected pipe is in the receiving direction, do not set the BVAL flag to 1.

33.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USB0.INTENB0 000A 0030h, USB1.INTENB0 000A 0230h

	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 always read as 0. The write value should always 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 function is selected; do not set these bits to 1 to enable the corresponding interrupt output when the host controller function is selected.

On detecting the interrupt corresponding to the bit in INTENB0 to which software has set 1, the USB generates the USB interrupt.

The USB sets 1 to each status bit in INTSTS0 when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in INTENB0 (regardless of whether the interrupt output is enabled or disabled).

While the status bit in INTSTS0 corresponding to the interrupt source indicates 1, the USB generates the USB interrupt when software modifies the corresponding interrupt enable bit in INTENB0 from 0 to 1.

33.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): USB0.INTENB1 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 always read as 0. The write value should always 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 always read as 0. The write value should always 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 always read as 0. The write value should always 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 INTENB1 can be set to 1 only when the host controller function is selected; do not set these bits to 1 to enable the corresponding interrupt output when the function controller function is selected.

INTENB1 specifies the interrupt masks when the host controller function is selected, and for the setup transaction.
On detecting the interrupt corresponding to the bit in INTENB1 to which software has set 1, the USB0 generates the USB interrupt.
The USB0 sets 1 to each status bit in INTSTS1 when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in INTENB1 (regardless of whether the interrupt output is enabled or disabled).
While the status bit in INTSTS1 corresponding to the interrupt source indicates 1, the USB0 generates the USB interrupt when software modifies the corresponding interrupt enable bit in INTENB1 from 0 to 1.
When the function controller function is selected, the interrupts should not be enabled.

33.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USB0.BRDYENB 000A 0036h, USB1.BRDYENB 000A 0236h

	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 always read as 0. The write value should always be 0.	R/W

BRDYENB enables or disables the INTSTS0.BRDY bit 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 BRDYENB to which software has set 1, the USB sets 1 to the corresponding PIPEnBRDY bit (n = 0 to 9) and the INTSTS0.BRDY bit, and generates the BRDY interrupt.

While at least one PIPEnBRDY bit indicates 1, the USB generates the BRDY interrupt when software modifies the corresponding interrupt enable bit in BRDYENB from 0 to 1.

33.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USB0.NRDYENB 000A 0038h, USB1.NRDYENB 000A 0238h

	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 always read as 0. The write value should always be 0.	R/W

NRDYENB enables or disables the INTSTS0.NRDY bit 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 NRDYENB to which software has set 1, the USB sets 1 to the corresponding PIPEnNRDY bit (n = 0 to 9) and the INTSTS0.NRDY bit, and generates the NRDY interrupt.

While at least one PIPEnNRDY bit indicates 1, the USB generates the NRDY interrupt when software modifies the corresponding interrupt enable bit in NRDYENB from 0 to 1.

33.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USB0.BEMPENB 000A 003Ah, USB1.BEMPENB 000A 023Ah

	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 always read as 0. The write value should always be 0.	R/W

BEMPENB enables or disables the INTSTS0.BEMP bit 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 BEMPENB to which software has set 1, the USB sets 1 to the corresponding PIPEnBEMP bit (n = 0 to 9) and the INTSTS0.BEMP bit, and generates the BEMP interrupt.

While at least one PIPEnBEMP bit in BEMPSTS indicates 1, the USB generates the BEMP interrupt when software modifies the corresponding interrupt enable bit in BEMPENB from 0 to 1.

33.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): USB0.SOFCFG 000A 003Ch, USB1.SOFCFG 000A 023Ch

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

Note 1. Bit b8 in the USB1.SOFCFG register is reserved.

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	EDGESTS *1	Edge Interrupt Output Status Monitor	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b8	TRNENSEL *1,*2	Transaction-Enabled Time Select	0: For non-low-speed communication 1: For low-speed communication	R/W
b15 to b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. This bit is unnecessary in a system which does not use the edge interrupt output signal.
Note 2. Bit b8 in the USB1.SOFCFG register is reserved. It is read as 0. When writing, write 0 to this bit.

EDGESTS Bit (Edge Interrupt Output Status Monitor)

The EDGESTS bit indicates 1 when the edge interrupt output signal is in the middle of the edge processing.
Make sure the EDGESTS bit is 0 when 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 USB0 issues tokens in a frame via the port.
Set the TRNENSEL bit to 1 when a low-speed device is connected via the HUB.
The TRNENSEL bit is valid only when the host controller function is selected.
This bit should be set to 0 if the function controller function is selected.

33.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): USB0.INTSTS0 000A 0040h, USB1.INTSTS0 000A 0240h

	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	0	0/1 *3	0	0	0	0

Note 1. This bit is initialized to 0 by a power-on reset and 1 by a USB bus reset.
Note 2. This bit is initialized to 1 when the level of the USBm_VBUS pin input is high and 0 when low.
Note 3. These bits are initialized to 000b by a power-on reset and 001b by a USB bus reset.

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage	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 1 1 1: Setting prohibited	R
b3	VALID	USB Request Reception	0: Not detected 1: Setup packet reception	R/W *1
b6 to b4	DVSQ[2:0]	Device State	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 x: Don't care	R
b7	VBSTS	VBUS Input Status	0: USBm_VBUS pin is low. 1: USBm_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status*2	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W *1
b12	DVST	Device State Transition Interrupt Status*2	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W *1
b13	SOFR	Frame Number Refresh Interrupt Status	0: SOF interrupts are not generated. 1: SOF interrupts are generated. (1) When the host controller function is selected The USB sets the SOFR bit to 1 on updating the frame number when software has set the UACT bit to 1. (A SOFR interrupt is detected every 1 ms.) (2) When the function controller function is selected The USB sets the SOFR bit to 1 on updating the frame number. (A SOFR 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.	R/W *1
b14	RESM	Resume Interrupt Status*2,*3	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W *1
b15	VBINT	VBUS Interrupt Status*3	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W *1

Note 1. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
Note 2. A change in the status of the RESM, DVST, and CTRT bits occur only when the host controller function is selected; disable the corresponding interrupt enable bits (set to 0) when the function controller function is selected.
Note 3. A change in the status indicated by the VBINT and RESM bits 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] Bits (Control Transfer Stage)

When the host controller function is selected, the read value is invalid.

VALID Bit (USB Request Reception)

When the host controller function is selected, the read value is invalid.

DVSQ[2:0] Bits (Device State)

The DVSQ[2:0] bits are initialized by a USB bus reset.
When the host controller function is selected, the read value is invalid.

BRDY Bit (Buffer Ready Interrupt Status)

Indicates the BRDY interrupt status.
The USB sets the BRDY bit to 1 when at least one PIPEnBRDY bit (n=0 to 9) is set to 1 among the PIPEBRDY bits corresponding to the 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 software enables the BRDY interrupt output).
For the conditions for PIPEnBRDY status assertion, refer to section 33.3.3.1, BRDY Interrupt.
The USB clears the BRDY bit to 0 when software writes 0 to all the PIPEnBRDY bits corresponding to the PIPEnBRDYE bits to which 1 has been set.
The BRDY bit cannot be cleared to 0 even if software writes 0 to this bit.

NRDY Bit (Buffer Not Ready Interrupt Status)

The USB sets the NRDY bit to 1 when at least one PIPEnNRDY bit (n=0 to 9) is set to 1 among the PIPENRDY bits 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 33.3.3.2, NRDY Interrupt.
The USB clears the NRDY bit to 0 when software writes 0 to all the PIPEnNRDY bits corresponding to the PIPEnNRDYE bits to which 1 has been set.
The NRDY bit cannot be cleared to 0 even if software writes 0 to this bit.

BEMP Bit (Buffer Empty Interrupt Status)

The USB sets the BEMP bit to 1 when at least one PIPEnBEMP bit (n=0 to 9) is set to 1 among the PIPEnBEMP bits 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 software enables the BEMP interrupt output).
For the conditions for PIPEnBEMP status assertion, refer to section 33.3.3.3, BEMP Interrupt.
The USB clears the BEMP bit to 0 when software writes 0 to all the PIPEnBEMP bits corresponding to the PIPEnBEMPE bits to which 1 has been set.
The BEMP bit cannot be cleared to 0 even if software writes 0 to this bit.

CTRT Bit (Control Transfer Stage Transition Interrupt Status)

When the function controller function is selected, the USB updates the value of the CTSQ[2:0] bits and sets the CTRT bit 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 function is selected, the read value is invalid.

DVST Bit (Device State Transition Interrupt Status)

When the function controller function is selected, the USB updates the DVSQ [2:0] value and sets the DVST bit 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 function is selected, the read value is invalid.

SOFR Bit (Frame Number Refresh Interrupt Status)

(1) When the host controller function is selected
The USB sets the SOFR bit to 1 on updating the frame number when software has set the UACT bit in DVSTCTR0 to 1. (A frame number refresh interrupt is detected every 1 ms.)
(2) When the function controller function is selected
The USB sets the SOFR bit 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 Bit (Resume Interrupt Status)

When the function controller function is selected, the USB sets the RESM bit to 1 on detecting the falling edge of the signal on the USBm_DP pin in the suspended state (DVSQ [2:0] = 1xxb).
When the host controller function is selected, the read value is invalid.

VBINT Bit (VBUS Interrupt Status)

The USB sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the USBm_VBUS pin input value. The USB sets the VBSTS bit to indicate the USBm_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS bit until the same value is read three or more times, and eliminate chattering.

33.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): USB0.INTSTS1 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 always read as 0. The write value should always be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status	0: SACK interrupts are not generated. 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status	0: SIGN interrupts are not generated. 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status	0: EOFERR interrupts are not generated. 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status	0: ATTCH interrupts are not generated. 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status*2	0: BCHG interrupts are not generated. 1: BCHG interrupts are generated.	R/W *1
b15	OVRCR	Overcurrent Input Change Interrupt Status*2	0: OVRCR interrupts are not generated. 1: OVRCR interrupts are generated.	R/W *1

Note 1. To clear the status indicated by the bits in INTSTS1, write 0 only to the bits to be cleared; write 1 to the other bits.
Note 2. A change in the status indicated by the OVRCR or BCHG bit 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 enabling the clock supply.
No interrupts other than those indicated by the BCHG and OVRCR bits can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

INTSTS1 is used to confirm the status of each interrupt when the host controller function is selected.
The various status change interrupts indicated by the bits in INTSTS1 should be enabled only when the host controller function is selected.

SACK Bit (Setup Transaction Normal Response Interrupt Status)

Indicates the status of the setup transaction normal response interrupt when the host controller function is selected.
The USB0 detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB0, and sets the SACK bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB0 generates the SACK interrupt.
When the function controller function is selected, the read value is invalid.

SIGN Bit (Setup Transaction Error Interrupt Status)

Indicates the status of the setup transaction error interrupt when the host controller function is selected.
The USB0 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 bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB0 generates the SIGN interrupt.
Specifically, the USB0 detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB0 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 function is selected, the read value is invalid.

EOFERR Bit (EOF Error Detection Interrupt Status)

Indicates the status of the EOFERR interrupt when the host controller function is selected.
The USB0 detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by the USB Specifications 2.0, and sets the EOFERR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB0 generates the EOFERR interrupt.
After detecting the EOFERR interrupt, the USB0 controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried for the USB port 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 function is selected, the read value is invalid.

ATTCH Bit (ATTCH Interrupt Status)

Indicates the status of the ATTCH interrupt when the host controller function is selected.
The USB0 detects the ATTCH interrupt on detecting J-state or K-state of the full-speed signal level for 2.5 µs, and sets the ATTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB0 generates the interrupt.
Specifically, the USB0 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 function is selected, the read value is invalid.

DTCH Bit (USB Disconnection Detection Interrupt Status)

Indicates the status of the USB disconnection detection interrupt when the host controller function is selected.
The USB0 detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB0 generates the interrupt.
The USB0 detects bus disconnection based on the USB Specifications 2.0.
After detecting the DTCH interrupt, the USB0 controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the USB port 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 function is selected, the read value is invalid.

BCHG Bit (USB Bus Change Interrupt Status)

Indicates the status of the USB bus change interrupt.
The USB0 detects the BCHG interrupt when a change in the full-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 bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB generates the interrupt.
The USB0 sets the LNST[1:0] bits 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] bits 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 function is selected, the read value is invalid.

OVRCR Bit (Overcurrent Input Change Interrupt Status)

Indicates the status of the USB0_OVRCURA and USB0_OVRCURB input pin change interrupt.
The USB0 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 bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB0 generates the interrupt.

33.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USB0.BRDYSTS 000A 0046h, USB1.BRDYSTS 000A 0246h

	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 for PIPE0*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status for PIPE1*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status for PIPE2*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status for PIPE3*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status for PIPE4*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status for PIPE5*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BRDY	BRDY Interrupt Status for PIPE6*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status for PIPE7*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status for PIPE8*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status for PIPE9*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the bits in BRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits.
Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

33.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USB0.NRDYSTS 000A 0048h, USB1.NRDYSTS 000A 0248h

	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 for PIPE0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status for PIPE1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status for PIPE2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status for PIPE3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status for PIPE4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status for PIPE5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status for PIPE6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status for PIPE7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status for PIPE8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status for PIPE9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. To clear the status indicated by the bits in NRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

33.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USB0.BEMPSTS 000A 004Ah, USB1.BEMPSTS 000A 024Ah

	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 for PIPE0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BEMP	BEMP Interrupt Status for PIPE1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BEMP	BEMP Interrupt Status for PIPE2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BEMP	BEMP Interrupt Status for PIPE3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BEMP	BEMP Interrupt Status for PIPE4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BEMP	BEMP Interrupt Status for PIPE5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BEMP	BEMP Interrupt Status for PIPE6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BEMP	BEMP Interrupt Status for PIPE7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BEMP	BEMP Interrupt Status for PIPE8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BEMP	BEMP Interrupt Status for PIPE9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. To clear the status indicated by the bits in BEMPSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

33.2.18 Frame Number Register (FRMNUM)

Address(es): USB0.FRMNUM 000A 004Ch, USB1.FRMNUM 000A 024Ch

	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	Frame number	R
b13 to b11	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b14	CRCE	Receive Data Error	0: No error 1: An error occurred	R/W *1
b15	OVRN	Overrun/Underrun Detection Status	0: No error 1: An error occurred	R/W *1

Note 1. Only 0 can be written.

FRNM[10:0] Bits (Frame Number)

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] bits at the SOF packet reception.
When reading these bits, read them twice and ensure that the values match.

CRCE Bit (Receive Data Error)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.
Software can clear the CRCE bit to 0 by writing 0 to the CRCE bit. Here, 1 should be written to the other bits in FRMNUM.
(1) When the host controller function is selected
On detecting a CRC error, the USB generates the internal NRDY interrupt request.
When the function controller function is selected
(2) On detecting a CRC error, the USB does not generate the internal NRDY interrupt request.

OVRN Bit (Overrun/Underrun Detection Status)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.
Software can clear the OVRN bit to 0 by writing 0 to the OVRN bit. Here, 1 should be written to the other bits in FRMNUM.
(1) When the host controller function is selected
The USB sets the OVRN bit 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 function is selected
The USB sets the OVRN bit 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.

33.2.19 Device State Change Register (DVCHGR)

Address(es): USB0.DVCHGR 000A 004Eh, USB1.DVCHGR 000A 024Eh

	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 always read as 0. The write value should always 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]. 1: Enables the writing to the USBADDR.STSRECOV[3:0] bits and USBADDR.USBADDR[6:0].	R/W

For details, see section 33.3.1.4, Release from Deep Software Standby Mode Due to USB Suspend/Resume Interrupts by a USB Suspend/Resume Interrupt.

33.2.20 USB Address Register (USBADDR)

Address(es): USB0.USBADDR 000A 0050h, USB1.USBADDR 000A 0250h

	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 function 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 always read as 0. The write value should always be 0.	R/W
b11 to b8	STSRECOV[3:0]	Status Recovery	• Recovery when the function controller function 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 (Configuared state) Other than above: Setting prohibited • Recovery when the host controller function is selected b11 b8 1 0 0 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) Other than above: Setting prohibited	R/W
b15 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

For details, see section 33.3.1.4, Release from Deep Software Standby Mode Due to USB Suspend/Resume Interrupts by a USB Suspend/Resume Interrupt.

USBADDR[6:0] Bits (USB Address)

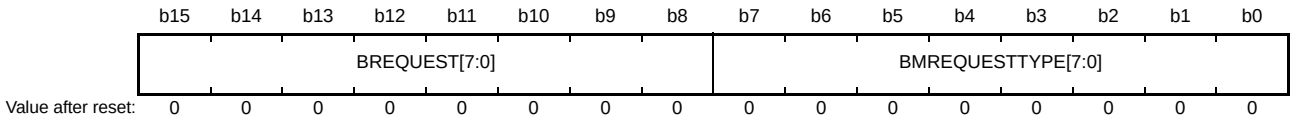
On detecting the USB bus reset, the USB sets the USBADDR[6:0] bits to 00h.
The writing to these bits is enabled while the DVCHGR.DVCHG bits are set to 1. On returning from the USB power shut-off, the operation can resume to the USB address before the shut-off by the software.
When the host controller function is selected, the USBADDR[6:0] bits are invalid.
The USBADDR[6:0] bits are initialized by a USB bus reset detection.

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, see section 33.3.1.4, 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.

33.2.21 USB Request Type Register (USBREQ)

Address(es): USB0.USBREQ 000A 0054h, USB1.USBREQ 000A 0254h



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 tob8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value.	R/W *1

Note 1. When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

USBREQ stores setup requests for control transfers.
When the function controller function is selected, the values of bRequest and bmRequestType that have been received are stored. When the host controller function is selected, the values of bRequest and bmRequestType to be transmitted are set.
USBREQ 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 host controller operation 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 function controller operation is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

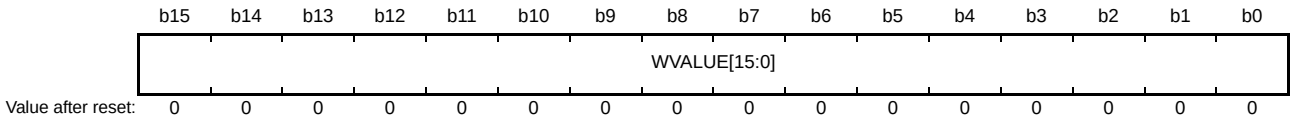
BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When host controller operation 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 function controller operation is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

33.2.22 USB Request Value Register (USBVAL)

Address(es): USB0.USBVAL 000A 0056h, USB1.USBVAL 000A 0256h



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 function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

When the function controller function is selected, the value of wValue that has been received is stored in USBVAL.
When the host controller function is selected, the value of wValue to be transmitted is set.
USBVAL is initialized by a USB bus reset.

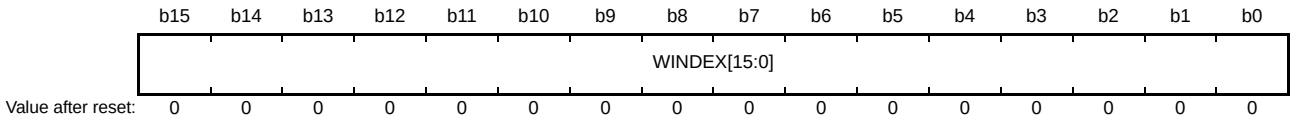
WVALUE[15:0] Bits (Value)

These bits store wRequest value of the USB request.

- When host controller operation 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 function controller operation 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.

33.2.23 USB Request Index Register (USBINDX)

Address(es): USB0.USBINDX 000A 0058h, USB1.USBINDX 000A 0258h



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 function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

USBINDX stores setup requests for control transfers.
When the function controller function is selected, the value of wIndex that has been received is stored. When the host controller function is selected, the value of wIndex to be transmitted is set.
USBINDX is initialized by a USB bus reset.

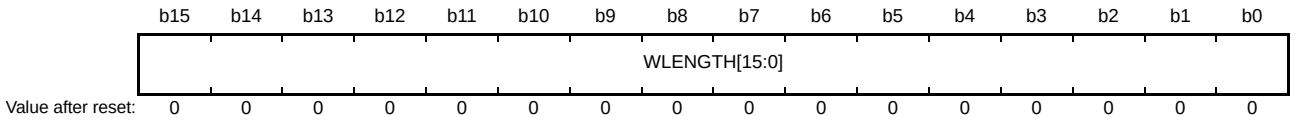
WINDEX[15:0] Bits (Index)

These bits hold the value of the wIndex field of a USB request.

- When host controller operation 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 DCPTR.SUREQ bit = 1.
- When function controller operation 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.

33.2.24 USB Request Length Register (USBLENG)

Address(es): USB0.USBLENG 000A 005Ah, USB1.USBLENG 000A 025Ah



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 function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

USBLENG stores setup requests for control transfers.
When the function controller function is selected, the value of wLength that has been received is stored. When the host controller function is selected, the value of wLength to be transmitted is set.
USBLENG 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 host controller operation 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 DCPTR.SUREQ bit = 1.
- When function controller operation 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.

33.2.25 DCP Configuration Register (DCPCFG)

Address(es): USB0.DCPCFG 000A 005Ch, USB1.DCPCFG 000A 025Ch

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

Note 1. Bits b4 in the USB1.DCPCFG register are reserved.

Bit	Symbol	Bit Name	Description	R/W
b3 TO b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	DIR*1	Transfer Direction*2	0: Data receiving direction 1: Data transmitting direction	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*2	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 always read as 0. The write value should always be 0.	R/W

Note 1. Bit b4 in the USB1.DCPCFG register is reserved. This bit is read as 0. When writing, write 0 to this bit.
Note 2. Modify this bit while PID is NAK. Before modifying this bit after modifying the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

DIR Bit (Transfer Direction)

When the host controller function is selected, the DIR bit sets the transfer direction of the data stage and status stage.
When the function controller function 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 to 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 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.

33.2.26 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USB0.DCPMAXP 000A 005Eh, USB1.DCPMAXP 000A 025Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]*1				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Note 1. Bits b15 to b12 in the USB1.DCPMAXP register are reserved.

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*2	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 always read as 0. The write value should always be 0.	R/W
b15 to b12	DEVSEL[3:0] *1	Device Select*3	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 Other than above: Setting prohibited	R/W

Note 1. Bits b15 to b12 in the USB1.DCPMAXP register are reserved. These bits are read as 0. When writing, write 0 to these bits.
Note 2. Modify the MXPS[6:0] bits while PID is NAK. Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB0, checking the PBUSY bit through software is not necessary. 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 3. Modify the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

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 the USB specification.
Do not write to the FIFO buffer or set PID = BUF while the setting of the MXPS[6:0] bits is 0.

DEVSEL[3:0] Bits (Device Select)

When the host controller function 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 0010, the address should be set to DEVADD2.
When the function controller function is selected, the DEVSEL[3:0] bits should be set to 0000b.

33.2.27 DCP Control Register (DCPCTR)

Address(es): USB0.DCPCTR 000A 0060h, USB1.DCPCTR 000A 0260h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ*1	—	—	SUREQCLR*1	—	—	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

Note 1. Bits b14 and b11 in the USB1.DCPCTR register are reserved.

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 always read as 0. The write value should always be 0.	R/W
b5	PBUSY	Pipe Busy	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*4	0: Invalid 1: Specifies DATA1.	R/W *2
b8	SQCLR	Toggle Bit Clear*4	0: Invalid 1: Specifies DATA0.	R/W *2
b10, b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b11	SUREQCLR*1	SUREQ Bit Clear	0: Invalid 1: Clears the SUREQ bit to 0.	R/W *3
b13, b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b14	SUREQ*1	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W *3
b15	BSTS	Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Bits b14 and b11 in the USB1.DCPCTR register are reserved. These bits are read as 0. When writing, write 0 to these bits.
Note 2. This bit is always read as 0.
Note 3. Only 1 can be written.
Note 4. Write 1 to the SQSET and SQCLR bits while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the DCP from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

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 function is selected

Modify the setting of the PID[1:0] bits from NAK to 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 is NAK, and then set PID to BUF. After PID has been set to 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 is NAK, and then set PID to BUF. After PID has been set to BUF, the USB executes the IN transaction.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID to STALL (11b) on receiving the data of a size exceeding the maximum packet size when software has set the PID[1:0] bits to BUF.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID to STALL (11b) on receiving the STALL handshake.

(2) When the function controller function is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to NAK on receiving the setup packet. Here, the USB sets the INTSTS0.VALID bit to 1. Software cannot modify the setting of the PID[1:0] bits until software sets the INTSTS0.VALID bit to 0.
- The USB sets PID to STALL (11b) on receiving the data of a size exceeding the maximum packet size when software has set the PID[1:0] bits to BUF.
- The USB sets PID to STALL (1xb) on detecting the control transfer sequence error.
- The USB sets PID to NAK on detecting the USB bus reset.

The USB does not check to 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 function is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.
When software sets the CCPL bit to 1 while the corresponding PID[1:0] bits are set to BUF, the USB completes the control transfer 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.
Software cannot write 1 to the CCPL bit while the INTSTS0.VALID bit is 1.
The PID[1:0] bits are initialized by a USB bus reset.
The CCPL bit is initialized by a USB bus reset.
When the host controller function is selected, be sure to write 0 to the CCPL bit.

PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from BUF to NAK.

The USB modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.

Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.

For details, refer to section 33.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Sequence Toggle Bit Monitor)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.

The USB allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

When the function controller function is selected, the USB sets the SQMON bit to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.

When the function controller function is selected, the USB does not reference the SQMON bit during the IN/OUT transaction of the status stage, and does not allow the SQMON bit to toggle upon normal completion.

SQSET Bit (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 (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 function is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit always 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 USB0 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 function 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 function is selected.

After completing the setup transaction process, the USB0 generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0.

The USB0 also clears 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, 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 NAK. After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, 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 function is selected, be sure to write 0 to the SUREQ bit.

BSTS Bit (Buffer Status)

Indicates whether DCP FIFO buffer access is enabled or disabled.

The meaning of the BSTS bit depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS bit indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS bit indicates whether the data to be transmitted can be written to the buffer.

33.2.28 Pipe Window Select Register (PIPESEL)

Address(es): USB0.PIPESEL 000A 0064h, USB1.PIPESEL 000A 0264h

	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 Other than above: Setting 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 PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.
After selecting the pipe using PIPESEL, functions of the pipe should be set using PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in PIPESEL.

PIPESEL[3:0] Bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number corresponding to 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 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 PIPECFG, PIPEMAXP, PIPEPERI, and PIPEnCTR. Writing to these bits is invalid.

33.2.29 Pipe Configuration Register (PIPECFG)

Address(es): USB0.PIPECFG 000A 0068h, USB1.PIPECFG 000A 0268h

	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 0000 means unused pipe.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always 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
b8	—	Reserved	This bit is always read as 0. The write value should always 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 always read as 0. The write value should always 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, SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID [1:0]bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

Note 2. Modify the BFRE, DBLB, and DIR bits while PID is 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 BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PIPEnCTR.PBUSY[1:0] bits 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 and CURPIPE bits are in the state described in the above note 2.

PIPECFG 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 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 (bits EPNUM[3:0] = 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 software has set the DIR bit to 0, 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 PID to 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 software has set the SHTNAK bit to 1 for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 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)

Specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.
When software has set the BFRE bit to 1 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, software needs to write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.
When software has set the BFRE bit to 1 and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.
For details, refer to section 33.3.3.1, 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 PID to 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.

33.2.30 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USB0.PIPEMAXP 000A 006Ch, USB1.PIPEMAXP 000A 026Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]*1				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0/1 *2	0	0	0	0	0	0

Note 1. Bits b15 to b12 in the USB1.PIPEMAXP register are reserved. These bits are always read as 0. The write value should always be 0.
Note 2. 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

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size bit*1	• 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 always read as 0. The write value should always be 0.	R/W
b15 to b12	DEVSEL[3:0] *1	Device Select bit*2	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 Other than above: Setting prohibited	R/W

Note 1. Modify the MXPS[8:0] bits while PID is 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 BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
Note 2. Modify the DEVSEL[3:0] bits while PID is NAK. To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB0, checking the PBUSY bit through software is not necessary.

PIPEMAXP specifies the maximum packet size for PIPE1 to PIPE9.

MXPS[8:0] Bits (Maximum Packet Size)

Specifies 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 the USB Specifications. While MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF.

DEVSEL[3:0] Bits (Device Select)

When the host controller function 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 function is selected, the DEVSEL[3:0] bits should be set to 0000b.

33.2.31 Pipe Cycle Control Register (PIPEPERI)

Address(es): USB0.PIPEPERI 000A 006Eh, USB1.PIPEPERI 000A 026Eh

	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] *1	Interval Error Detection Interval	Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2.	R/W
b11 to b3	—	Reserved	These bits are always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W

Note 1. Modify the IITV[2:0] bits while PID is NAK. To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

PIPEPERI 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 PID to 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 PIPE.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.
When the function controller function 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 function 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.

33.2.32 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Address(es): USB0.PIPE1CTR 000A 0070h, USB0.PIPE2CTR 000A 0072h, USB0.PIPE3CTR 000A 0074h, USB0.PIPE4CTR 000A 0076h, USB0.PIPE5CTR 000A 0078h, USB1.PIPE1CTR 000A 0270h, USB1.PIPE2CTR 000A 0272h, USB1.PIPE3CTR 000A 0274h, USB1.PIPE4CTR 000A 0276h, USB1.PIPE5CTR 000A 0278h

	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 always read as 0. The write value should always be 0.	R/W
b5	PBUSY	Pipe Busy	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	Toggle Bit Clear*2	0: Invalid 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 always read as 0. The write value should always be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor	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	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 PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
Note 3. Modify the ACLRM bit while PID[1:0] is 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 BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

PIPEnCTR can be set regardless of the pipe selection in PIPESEL.

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 is NAK. Modify the setting of the PID[1:0] bits to BUF to use the relevant pipe for USB transfer. Table 33.6 and Table 33.7 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 BUF to NAK during USB communication using the relevant pipe, check that the PBUSY bit is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the PIPECFG.SHTNAK bit for the selected pipe to 1.
- The USB sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets PID to NAK on detecting a USB bus reset when the function controller function is selected.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller function is selected.
- The USB sets PID to STALL (11) on receiving the STALL handshake when the host controller function 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 to BUF, set 00b (NAK) and then 01b (BUF).

PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.
Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.
For details, refer to section 33.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Toggle Bit Confirmation)

The SQMON bit 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 bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

SQSET Bit (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 (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 33.8 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 function 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 = 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] bit = 01b and the PIPECFG.DIR bit = 0)
When the ATREPM bit = 1 and PID = 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 function is selected, be sure to set the ATREPM bit to 0.

INBUFM Bit (Transmit Buffer Monitor)

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 bit to 1 when software (or DMAC) completes writing data to at least one FIFO buffer plane.
The USB sets the INBUFM bit 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 bit to 0 when the USB completes transmitting the data from the two FIFO buffer planes before software (or DMAC) completes writing data to one FIFO buffer plane.
The INBUFM bit indicates the same value as the BSTS bit when the relevant pipe is in the receiving direction (the PIPECFG.DIR bit = 0).

BSTS Bit (Buffer Status)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS bit depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 33.9.

Table 33.6 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller Function is Selected)

Bits PID[1:0]	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 33.7 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller Function is Selected)

Bits PID[1:0]	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. For the operation when the ATREPM bit is 1, refer to the description of the ATREPM bit.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (the 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 (the 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 (the 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 (the 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 (the 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 33.8 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 33.9 Operation of BSTS Bit

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Bit 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. Software has set the BCLR bit in the port control register to 1 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): USB0.PIPE6CTR 000A 007Ah, USB0.PIPE7CTR 000A 007Ch, USB0.PIPE8CTR 000A 007Eh, USB0.PIPE9CTR OOOA 0080h
USB1.PIPE6CTR 000A 027Ah, USB1.PIPE7CTR 000A 027Ch, USB1.PIPE8CTR 000A 027Eh, USB1.PIPE9CTR OOOA 0280h

	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 always read as 0. The write value should always be 0.	R/W
b5	PBUSY	Pipe Busy	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	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 always read as 0. The write value should always be 0.	R/W
b15	BSTS	Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read.
Note 2. Write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
Note 3. Modify the ACLRM bit while PID is 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 BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

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 is NAK. Modify the setting of the PID[1:0] bits to BUF to use the relevant pipe for USB transfer. Table 33.6 and Table 33.7 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 BUF to NAK during USB communication using the relevant pipe, check that the PBUSY bit is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the PID bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets PID to NAK on detecting a USB bus reset when the function controller function is selected.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller function is selected.
- The USB sets PID to STALL (11b) on receiving the STALL handshake when the host controller function 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 to BUF, set 00b (NAK) and then 01b (BUF).

PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.
Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.

SQMON Bit (Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
The USB allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

SQSET Bit (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 (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 33.10 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 Bit (Buffer Status)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS bit depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 33.9.

Table 33.10 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 function 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

33.2.33 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5)

Address(es): USB0.PIPE1TRE 000A 0090h, USB0.PIPE2TRE 000A 0094h, USB0.PIPE3TRE 000A 0098h, USB0.PIPE4TRE 000A 009Ch, USB0.PIPE5TRE 000A 00A0h, USB1.PIPE1TRE 000A 0290h, USB1.PIPE2TRE 000A 0294h, USB1.PIPE3TRE 000A 0298h, USB1.PIPE4TRE 000A 029Ch, USB1.PIPE5TRE 000A 02A0h

	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 always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W

Note: • Modify each bit in PIPEnTRE while PID is NAK. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

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 bits to 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 out the last received data.

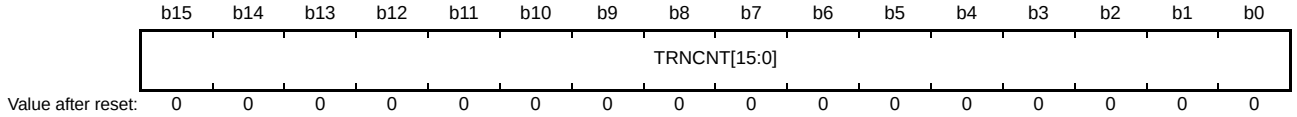
For the pipe in the transmitting direction, set the TRENB bit to 0.

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.

33.2.34 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5)

Address(es): USB0.PIPE1TRN 000A 0092h, USB0.PIPE2TRN 000A 0096h, USB0.PIPE3TRN 000A 009Ah, USB0.PIPE4TRN 000A 009Eh, USB0.PIPE5TRN 000A 00A2h, USB1.PIPE1TRN 000A 0292h, USB1.PIPE2TRN 000A 0296h, USB1.PIPE3TRN 000A 029Ah, USB1.PIPE4TRN 000A 029Eh, USB1.PIPE5TRN 000A 02A2h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TRNCNT[15:0]	Transaction Counter	• When written to: Specifies the number of transactions to be transferred through DMA. • 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

PIPEnTRN 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 clears 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
- Software has set the PIPEnTRE.TRCLR bit to 1.

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.

33.2.35 Device Address n Configuration Registers (DEVADDn) (n = 0 to 5)

Address(es): USB0.DEVADD0 000A 00D0h, USB0.DEVADD1 000A 00D2h, USB0.DEVADD2 000A 00D4h, USB0.DEVADD3 000A 00D6h, USB0.DEVADD4 000A 00D8h, USB0.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 always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W

DEVADDn specifies the transfer speed of the peripheral device which is the communication target for PIPE0 to PIPE9. When the host controller function is selected, the bits in DEVADDn should be set before starting communication using each pipe.
The bits in DEVADDn 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:

- DEVADDn is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to 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)

Specifies the USB transfer speed of the communication target peripheral device.
Set these bits to 01b when a low-speed device is connected via the HUB, whereas set them to 10b when a full-speed device is connected.
When the host controller function is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.
When the function controller function is selected, set these bits to 00b.

33.2.36 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
	DVBSTS1	—	DOVCB1	DOVCA1	—	—	DM1	DP1	DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
Value after reset:	x	0	x	x	0	0	x	x	x	0	x	x	0	0	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FIXPHY1	—	—	—	SRPC1	—	—	—	FIXPHY0	—	—	—	SRPC0
Value after 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	SRPC0	USB0 Single End Receiver Control	0: Input through the DP and DM inputs is disabled. 1: Input through the DP and DM inputs is enabled.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	FIXPHY0	USB0 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
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b8	SRPC1	USB1 Single End Receiver Control	0: Input through DP and DM is disabled. 1: Input through DP and DM is enabled.	R/W
b11 to b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b12	FIXPHY1	USB1 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 b13	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b16	DP0	USB0 DP Input	Indicates the DP input signal on USB port 0.	R
b17	DM0	USB0 DM Input	Indicates the DM input signal on USB port 0.	R
b19, b18	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b20	DOVCA0	USB0 OVRCURA Input	Indicates the OVRCURA input signal on USB port 0.	R
b21	DOVCB0	USB0 OVRCURB Input	Indicates the OVRCURB input signal on USB port 0.	R
b22	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b23	DVBSTS0	USB0 VBUS Input	Indicates the VBUS input signal on USB port 0.	R
b24	DP1	USB1 DP Input	Indicates the DP input signal on USB port 1.	R
b25	DM1	USB1 DM Input	Indicates the DM input signal on USB port 1.	R
b27, b26	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b29, b28	—	Reserved	These bits are always read as undefined. Writing to these bits has no effect.	R
b30	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b31	DVBSTS1	USB1 VBUS Input	Indicates the VBUS input signal on USB port 1.	R

SRPCn Bits (USBn Single End Receiver Control) (n = 0 or 1)

Each SRPCn bit controls the DP and DM inputs of the transceiver for USB port n.

Each bit is only valid when the FIXPHYn bit is 1.

FIXPHYn Bits (USBn Transceiver Output Fix) (n = 0 or 1)

Each FIXPHYn bit keeps the outputs for the transceiver on USB port n disabled.

33.2.37 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
	DVBINT1	—	DOVRCRB1	DOVRCRA1	—	—	DMINT1	DPINT1	DVBINT0	—	OVRCURBIN	OVRCURAIN	—	—	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
	DVBSE1	—	DOVRCRBE1	DOVRCRAE1	—	—	DMINTE1	DPINTE1	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	USB0 DP Interrupt Enable/Clear	0: Triggering of recovery from deep software standby mode by DP input to USB port 0 is disabled. 1: Triggering of recovery from deep software standby mode by DP input to USB port 0 DP input is enabled.	R/W
b1	DMINTE0	USB0 DM Interrupt Enable/Clear	0: Triggering of recovery from deep software standby mode by DM input to USB port 0 is disabled. 1: Triggering of recovery from deep software standby mode by DM input to USB port 0 is enabled.	R/W
b3, b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	DOVRCRAE0	USB0 OVRCURA Interrupt Enable/Clear	0: Triggering of recovery from deep software standby mode by OVRCURA input to USB port 0 is disabled. 1: Triggering of recovery from deep software standby mode by OVRCURA input to USB port 0 is enabled.	R/W
b5	DOVRCRBE0	USB0 OVRCURB Interrupt Enable/Clear	0: Triggering of recovery from deep software standby mode by OVRCURB input to USB port 0 is disabled. 1: Triggering of recovery from deep software standby mode by OVRCURB input to USB port 0 is enabled.	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	DVBSE0	USB0 VBUS Interrupt Enable/Clear	0: Triggering of recovery from deep software standby mode by VBUS input to USB port 0 is disabled. 1: Triggering of recovery from deep software standby mode by VBUS input to USB port 0 is enabled.	R/W
b8	DPINTE1	USB1 DP Interrupt Enable/Clear	0: Triggering of recovery from deep software standby mode by DP input to USB port 1 is disabled. 1: Triggering of recovery from deep software standby mode by DP input to USB port 1 is enabled.	R/W
b9	DMINTE1	USB1 DM Interrupt Enable/Clear	0: Triggering of recovery from deep software standby mode by DM input to USB port 1 is disabled. 1: Triggering of recovery from deep software standby mode by DM input to USB port 1 DM input is enabled.	R/W
b11 to b14	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15	DVBSE1	USB1 VBUS Interrupt Enable/Clear	0: Triggering of recovery from deep software standby mode by the USB port 1 VBUS input is disabled. 1: Triggering of recovery from deep software standby mode by the USB port 1 VBUS input is enabled.	R/W
b16	DPINT0	USB0 DP Interrupt Source Recovery	0: The system has not returned from deep software standby mode due to DP input to USB port 0. 1: The system has returned from deep software standby mode due to DP input to USB port 0.	R
b17	DMINT0	USB0 DM Interrupt Source Recovery	0: The system has not returned from deep software standby mode due to DM input to USB port 0. 1: The system has returned from deep software standby mode due to DM input to USB port 0.	R
b19, 18	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	OVRCURAINT0	USB0 OVRCURA Interrupt Source Recovery	0: The system has not returned from deep software standby mode due to OVRCURA input to USB port 0. 1: The system has returned from deep software standby mode due to OVRCURA input to USB port 0.	R
b21	OVRCURBINT0	USB0 OVRCURB Interrupt Source Recovery	0: The system has not returned from deep software standby mode by the USB port 0 OVRCURB input. 1: The system has returned from deep software standby mode by the USB port 0 OVRCURB input.	R
b22	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b23	DVBINT0	USB0 VBUS Interrupt Source Recovery	0: The system has not returned from deep software standby mode due to VBUS input to USB port 0. 1: The system has returned from deep software standby mode due to VBUS input to USB port 0.	R
b24	DPINT1	USB1 DP Interrupt Source Recovery	0: The system has not returned from deep software standby mode due to DP input to USB port 1. 1: The system has returned from deep software standby mode by USB port 1 DP input.	R
b25	DMINT1	USB1 DM Interrupt Source Recovery	0: The system has not returned from deep software standby mode due to DM input to USB port 1. 1: The system has returned from deep software standby mode due to DM input to USB port 1.	R
b27, b26	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b29, b28	—	Reserved	These bits are always read as 0. The write value should always be 0.	R
b30	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b31	DVBINT1	USB1 VBUS Interrupt Source Recovery	0: The system has not returned from deep software standby mode due to VBUS input to USB port 1. 1: The system has returned from deep software standby mode due to VBUS input to USB port 1.	R

DPINTEn Bit (USBn DP Interrupt Enable/Clear) (n = 0 or 1)

Each DPINTEn bit enables or disables triggering of recovery from deep software standby mode by the DP input to USB port n. Writing 0 to the bit while the DPINTn bit is set to 1 clears DPINTn to 0.

DMINTEn Bit (USBn DM Interrupt Enable/Clear) (n = 0 or 1)

Each DMINTENn bit enables or disables triggering of recovery from deep software standby mode by the DM input to USB port n. Writing 0 to the bit while the DMINTn bit is set to 1 clears DMINTn to 0.

DOVRCRAEn Bit (USBn OVRCURA Interrupt Enable/Clear) (n = 0 or 1)

This bit enables or disables triggering of recovery from deep software standby mode by the OVRCURA input to USB port 0. Writing 0 to the bit while the DOVRCRA0 bit is set to 1 clears the DOVRCRA0 bit to 0.

DOVRCRBEn Bit (USBn OVRCURB Interrupt Enable/Clear) (n = 0 or 1)

This bit enables or disables triggering of recovery from deep software standby mode by the OVRCURB input to USB port 0. Writing 0 to the bit while the DOVRCRB0 bit is set to 1 clears the DOVRCRB0 bit to 0.

DVBSEn Bit (USBn VBUS Interrupt Enable/Clear) (n = 0 or 1)

Each DVBSEn bit enables or disables triggering of recovery from deep software standby mode by the VBUS input to USB port n. Writing 0 to the bit while the DVBINTn bit is set to 1 clears the DVBINTn bit to 0.

DPINTn Bit (USBn DP Interrupt Source Recovery) (n = 0 or 1)

Each DPINTn bit indicates that the system has returned from deep software standby mode due to the DP input to USB

port n. Recovery from deep software standby mode due to DP input to USBn is only enabled when the DPINTEn bit is 1.
When the DPINTn bit is set to 1, writing 0 to the DPINTEn bit clears the DPINTn bit to 0.

DMINTn Bit (USBn DM Interrupt Source Recovery) (n = 0 or 1)

The DMINTn bit indicates that the system has returned from deep software standby mode due to the DM input to USB port n. Recovery from deep software standby mode due to DM input to USBn is only enabled when the DMINTEn bit is 1.
When the DMINTn bit is set to 1, writing 0 to the DMINTEn bit clears the DMINTn bit to 0.

DOVRCRA0 Bit (USB0 OVRCURA Interrupt Source Recovery)

This bit indicates that the system has returned from deep software standby mode due to the OVRCURA input to USB port 0. Recovery from deep software standby mode due to OVRCURA input to USB port 0 is only enabled when the DOVRCRAE0 bit is 1.
When the DOVRCRA0 bit is set to 1, writing 0 to the DOVRCRAE0 bit clears the DOVRCRA0 bit to 0.

DOVRCRB0 Bit (USB0 OVRCURB Interrupt Source Recovery)

This bit indicates that the system has returned from deep software standby mode due to the OVRCURB input to USB port 0. Recovery from deep software standby mode due to OVRCURB input to USB port 0 is only enabled when the DOVRCRBE0 bit is 1.
When the DOVRCRB0 bit is set to 1, writing 0 to the DOVRCRBE0 bit clears the DOVRCRB0 bit to 0.

DVBINTn Bit (USBn VBUS Interrupt Source Recovery) (n = 0 or 1)

The DVBINTn bit indicates that the system has returned from deep software standby mode due to the VBUS input to USB port n. Recovery from deep software standby mode due to VBUS input to USB port n is only enabled when the DVBSEn bit is 1.
When the DVBINTn bit is set to 1, writing 0 to the DVBSEn bit clears the DVBINTn bit to 0.

33.3 Operation

33.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

33.3.1.1 Starting Operation

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (the SYSCFG.SCKE bit = 1) enables and starts USB operation.

33.3.1.2 Controller Function Selection

For the USB0, the host controller function or function controller function can be selected using the DCFM bit in SYSCFG. The DCFM bit should be modified in the initial settings immediately after a power-on reset or in the D+ pull-up-disabled (the SYSCFG.DPRPU bit = 0) and D+ and D- pull-down-disabled (the SYSCFG.DRPD bit = 0) state.

33.3.1.3 Example of USB External Connection Circuit

Figure 33.2 shows an example of OTG connection of the USB connector (USB0) in the self-powered state.
The USB0 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 function is selected and the DPRPU bit is cleared 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.

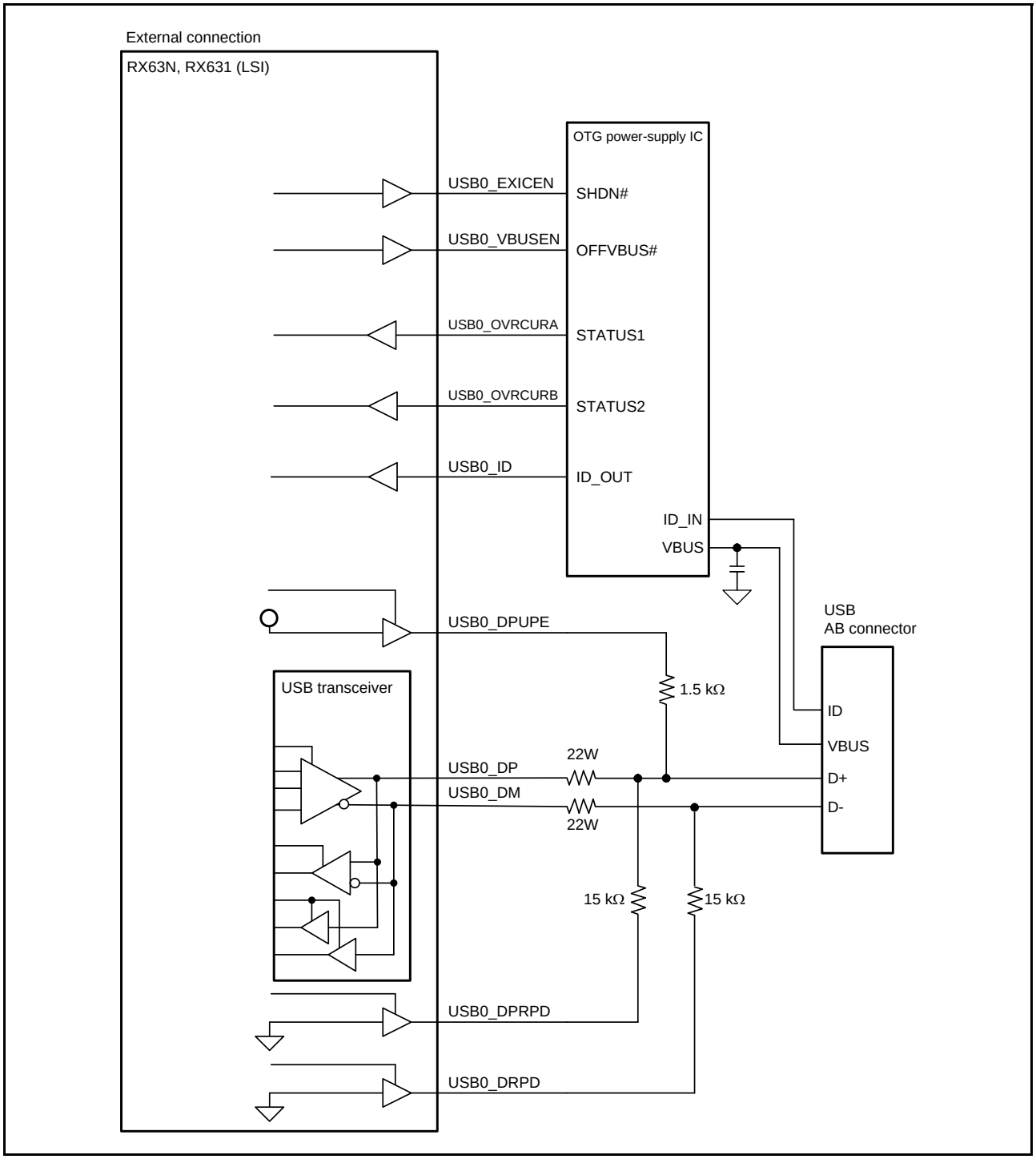


Figure 33.2 Sample OTG Connection of USB Connector (USB0) in Self-Powered State

Figure 33.3 shows an example of functional connection sample of the USB connector in the self-powered state.

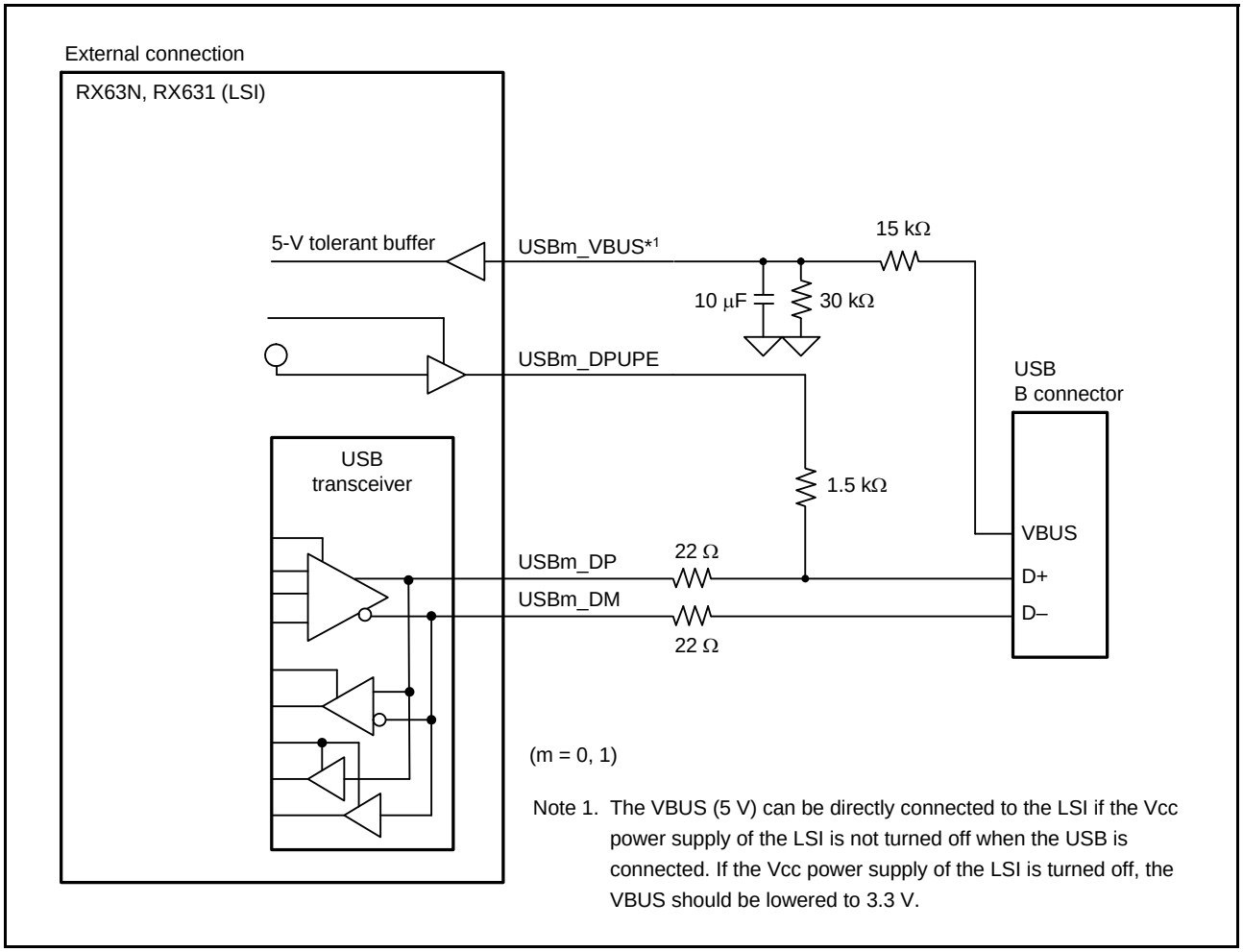


Figure 33.3 Functional Connection Sample of USB Connector in Self-Powered State

Figure 33.4 shows an example of host connection of the USB connector (USB0).

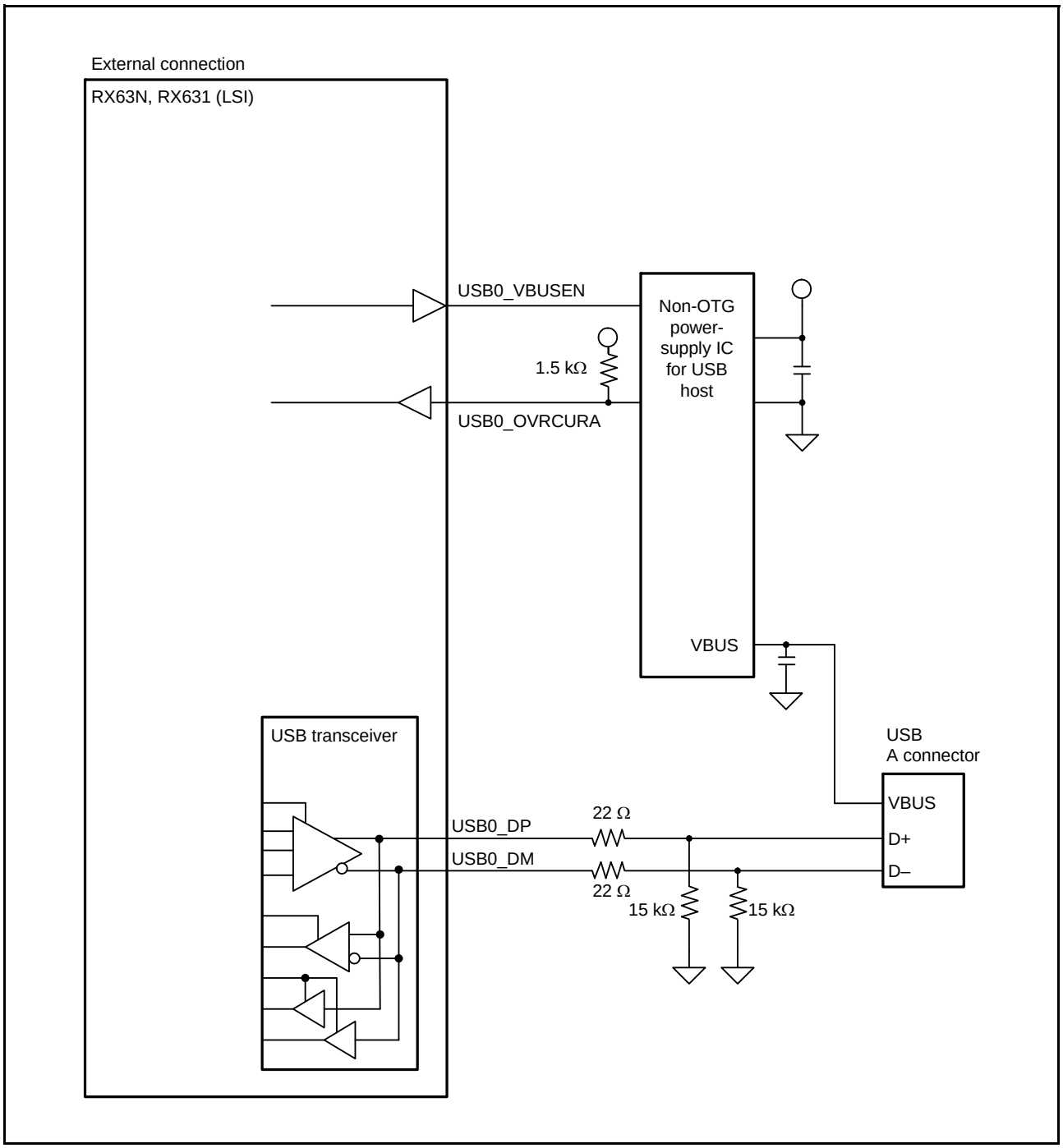


Figure 33.4 Sample Host Connection of USB Connector (USB0)

Figure 33.5 shows a functional connection example of the USB connector in the bus-powered state.

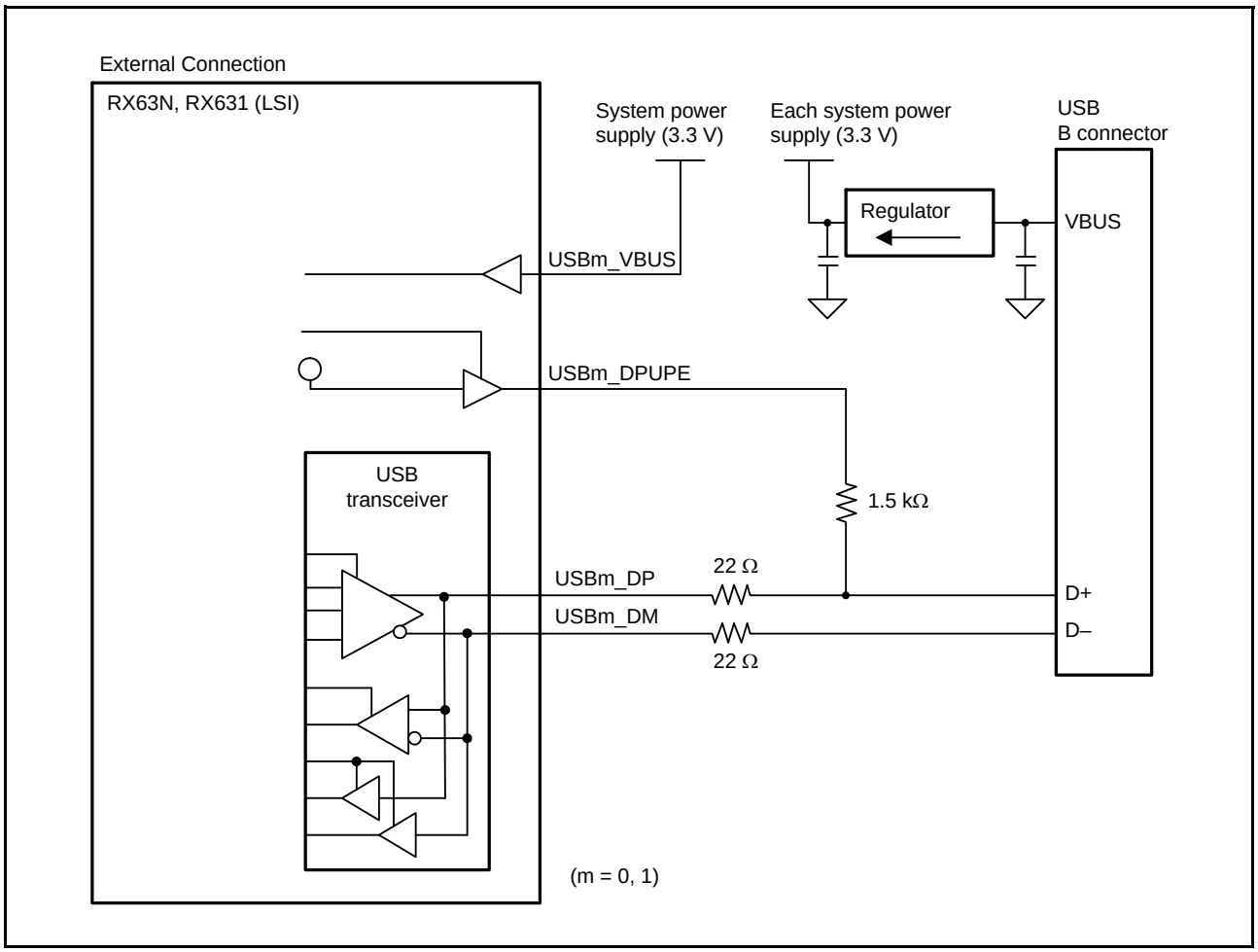


Figure 33.5 Functional Connection Example of USB Connector in Bus-Powered State

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

33.3.1.4 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 USB0 and USB1 to detect USB suspend/resume interrupts.
Figure 33.6 shows a schematic diagram of connection between the USB resume detecting unit and the I/O pins for USB0. Figure 33.7 shows a schematic diagram of connection between the USB resume detecting unit and the I/O pins for USB1.

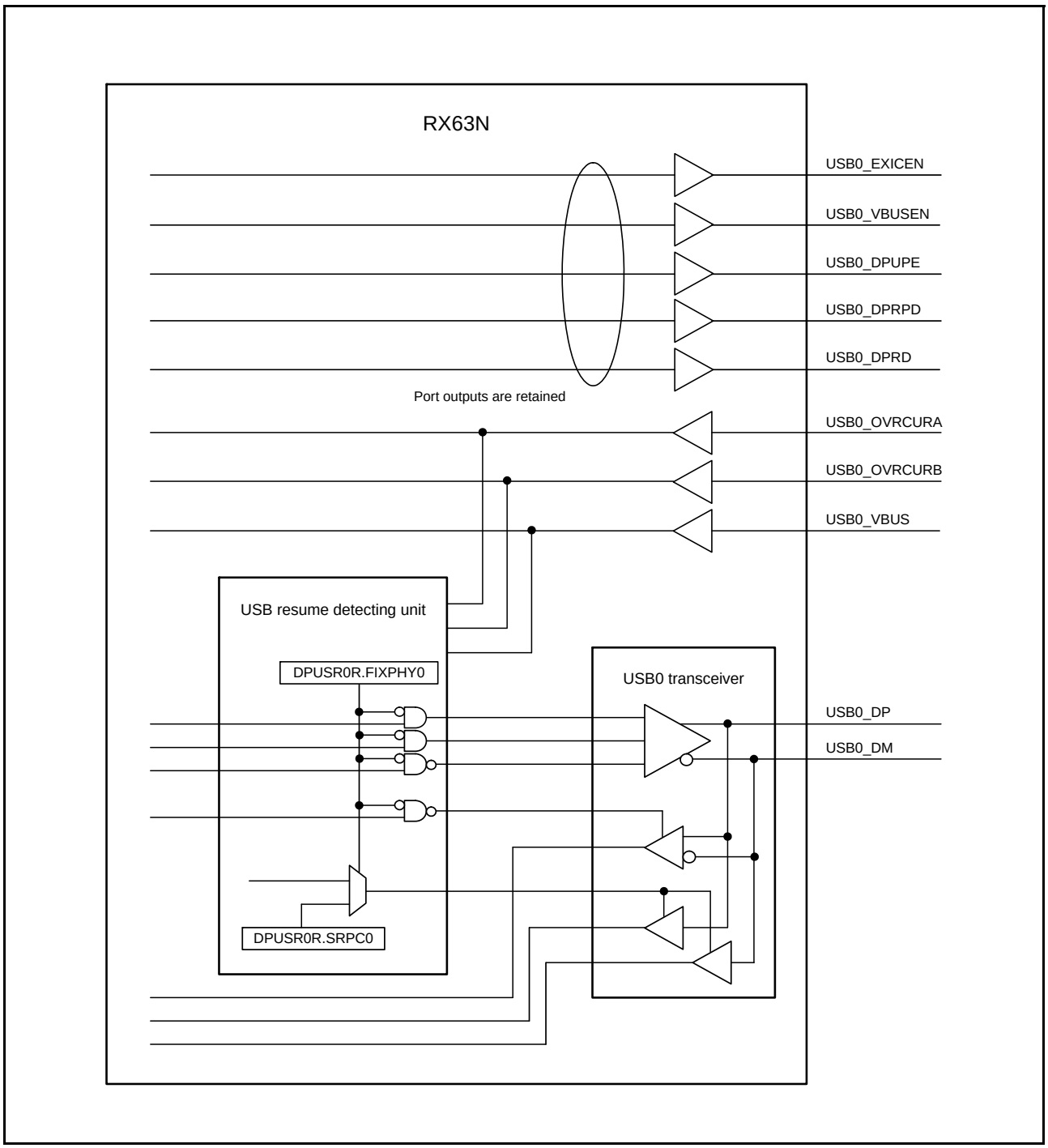


Figure 33.6 Overview of Connection between USB Resume Detecting Unit and USB0 I/O Pins

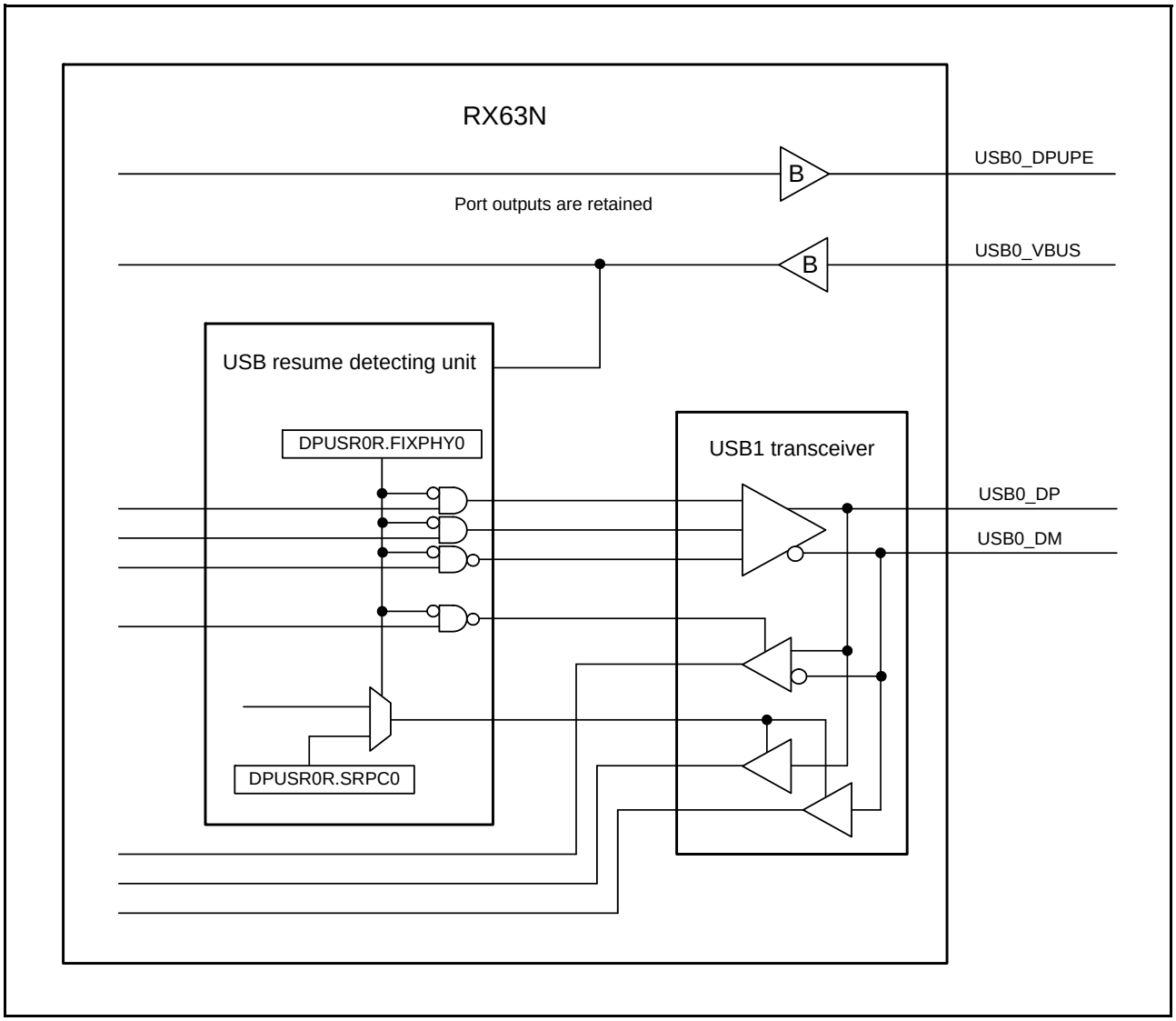


Figure 33.7 Overview of Connection between USB Resume Detecting Unit and USB1 I/O Pins

Table 33.11 shows the USB suspend/resume interrupt sources and their corresponding I/O pins.

Table 33.11 USB Suspend/Resume Interrupt Sources and Corresponding I/O Pins

Port	USB Operating Mode	Source	Pin Name
USB0	Function/OTG	Resume	USB0_DP
	Host/OTG	Connection/Disconnection	USB0_DP/USB0_DM
	Function	Connection/Disconnection	USB0_VBUS
	Host	Overcurrent detection	USB0_OVRCURA
	OTG mode	Overcurrent detection	USB0_OVRCURA/USB0_OVRCURB
USB1	Function	Resume	USB1_DP
		Connection/Disconnection	USB1_VBUS

When canceling deep software standby mode using a USB suspend/resume interrupt, set the DPSBYCR.IOKEEP bit to have the outputs of the I/O ports retained.

Figure 33.8 shows a flowchart for setting the USB when entering deep software standby mode while the host controller function or function controller function is selected. Figure 33.9 shows a flowchart for setting the USB when canceling deep software standby mode while the host controller function is selected. Figure 33.10 shows a flowchart for setting the USB when canceling deep software standby mode while the function controller function is selected.

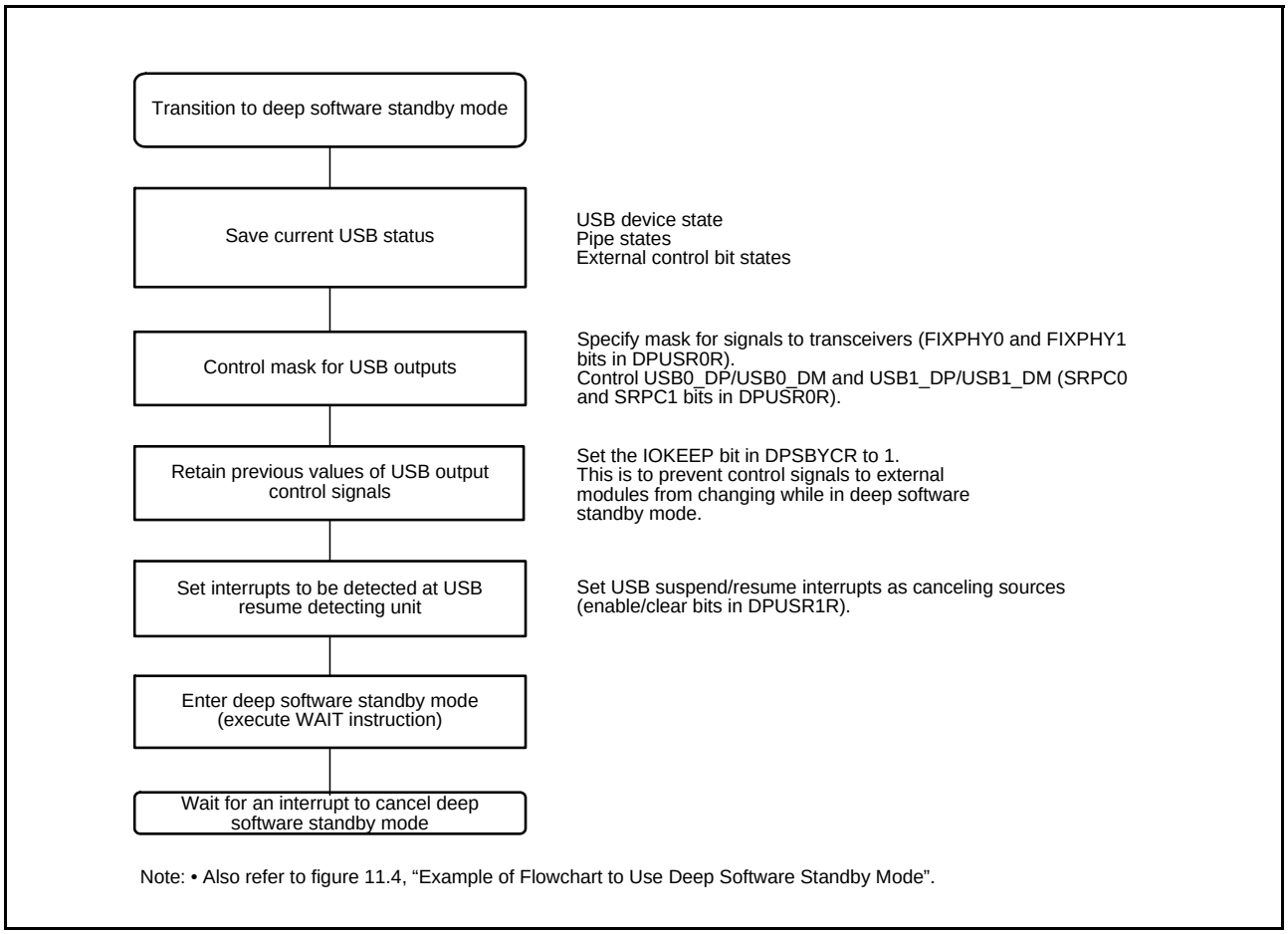


Figure 33.8 USB Setting Flowchart for Transition to Deep Software Standby Mode as Host/Function Controller

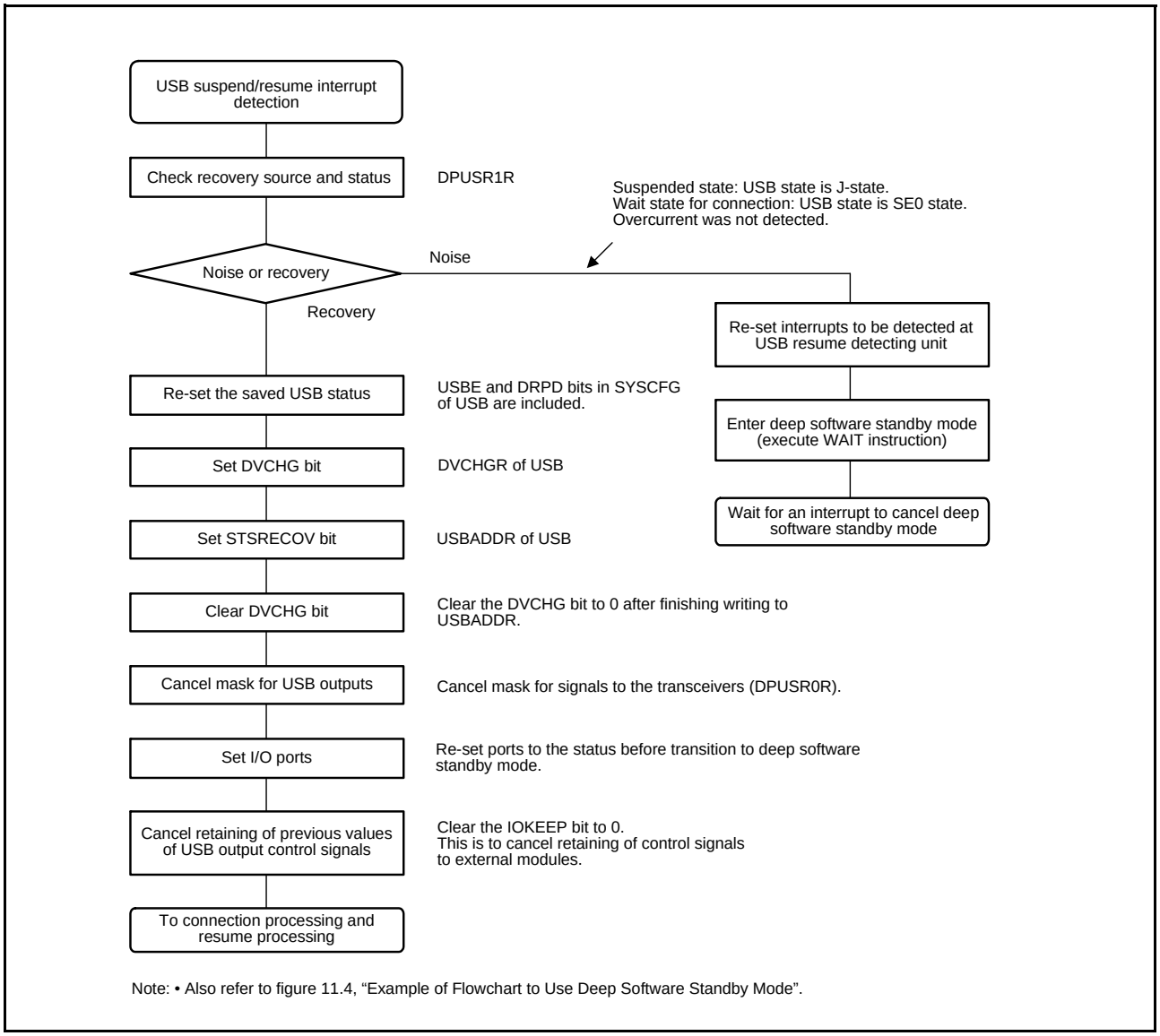


Figure 33.9 USB Setting Flowchart for Canceling Deep Software Standby Mode as Host Controller

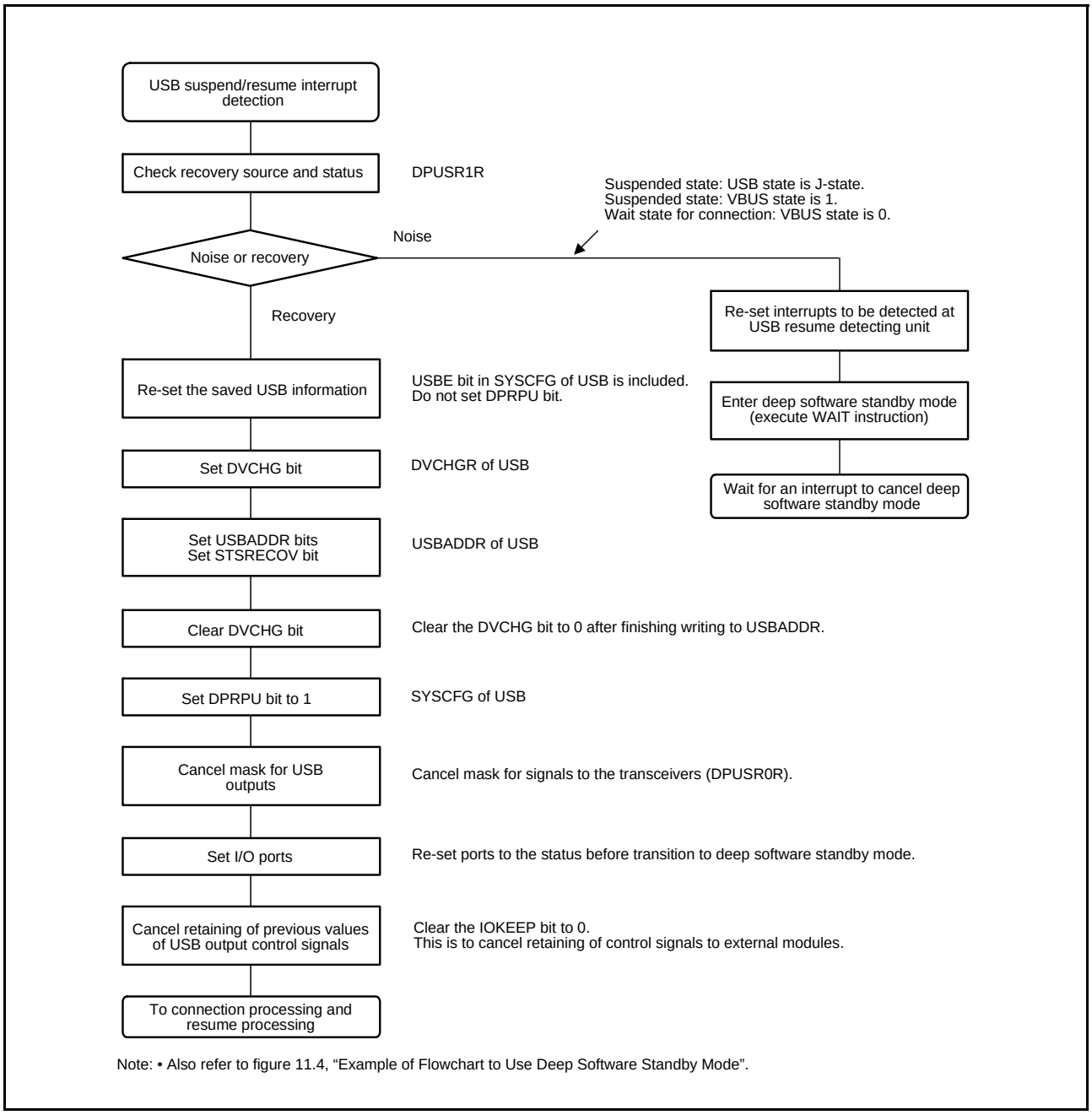


Figure 33.10 USB Setting Flowchart for Canceling Deep Software Standby Mode as Function Controller

33.3.2 Interrupt Sources

Table 33.12 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, the USB issues a USB interrupt request to the interrupt controller (ICUA) and an USB interrupt will be generated.

Table 33.12 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 USBm_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 function is selected] • When an SOF packet with a different frame number has been transmitted [Function controller function 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 function 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 function is selected] • When NAK has been returned for an IN or OUT token while the PID bit = 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*2	• 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*2	• When a change of USB bus state has been detected	Host/function	SYSSTS0. LNST[1:0]
DTCH	Disconnection detection during full-speed operation*2	• When disconnection of a peripheral device has been detected in full-speed operation	Host	DVSTCTR0. RHST[2:0]
ATTCH	Device connection detection*2	• When J-state or K-state is detected on the USB port for 2.5 s. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection*2	• When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal setup operation*2	• When the normal response (ACK) for the setup transaction has been received	Host	—
SIGN	Setup error*2	• 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.
Note 2. These interrupts are not provided in USB1.

Figure 33.11 and Figure 33.12 show the circuits related to the interrupts in the USB0 and USB1, respectively.

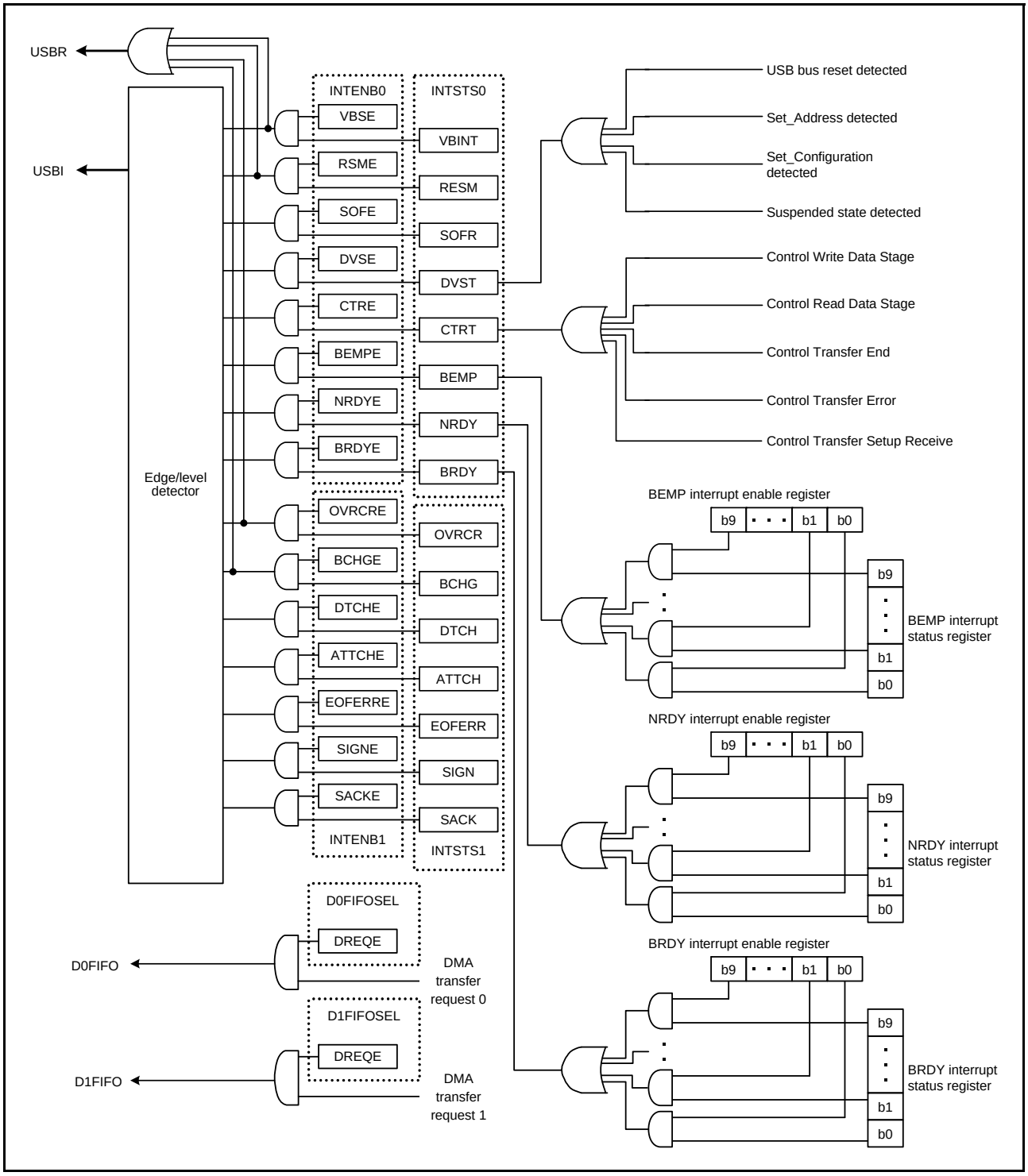


Figure 33.11 Circuits Related to Interrupts in the USB0

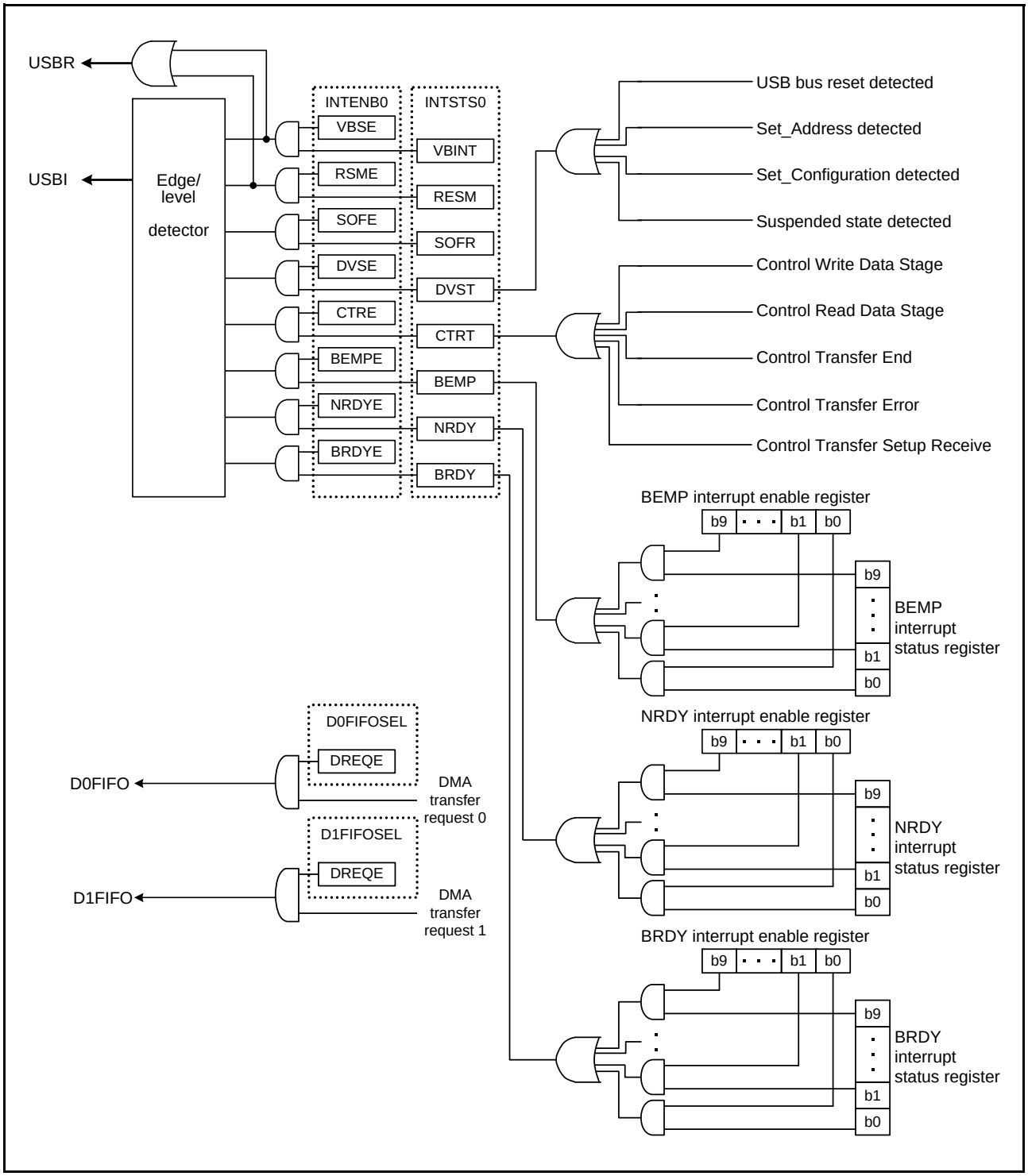


Figure 33.12 Circuits Related to Interrupts in the USB1

Table 33.13 shows the interrupts generated in the USBm (m = 0 or 1).

Table 33.13 USBm Interrupts

Interrupt Name	Interrupt Flag	DTC Activation	DMAC Activation	Priority
D0FIFO	DMA transfer request 0	Possible	Possible	High
D1FIFO	DMA 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 *1, bus change interrupt*1, disconnection detection during full-speed operation*1, device connection detection*1, EOF error detection*1, normal setup operation*1, and setup error*1	Not possible	Not possible	Low
USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, and bus change interrupt	Not possible	Not possible	—

Note 1. These interrupts are not provided in USB1.

33.3.3 Interrupt Descriptions

33.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller function or function controller function is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in BRDYSTS. Under this condition, the USB generates a BRDY interrupt if software has set 1 to the 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 bit corresponding to the pertinent pipe.

(a) For the pipe in the transmitting direction:

- When software changes the DIR bit from 0 to 1.
- 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 bit 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 bit is read as 0).
  No request trigger is generated for the transaction in which DATA-PID disagreement 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 function is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEnBRDY bit in BRDYSTS through software. In this case, 1s should be written to the PIPEBRDY bits 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 BRDYSTS 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 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 out after any of the above conditions has been satisfied, the USB determines that all data for a single transfer has been completely read out.
When a zero-length packet is received while the FIFO buffer is empty, the USB determines that all data for a single transfer has been completely read out upon passing the zero-length packet data to the CPU. 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 PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY bit through software. In this case, 1s should be written to the PIPEBRDY 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 = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) 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 bit cannot be cleared to 0 through software.
When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be cleared to 0.

Figure 33.13 shows the timing of BRDY interrupt generation.

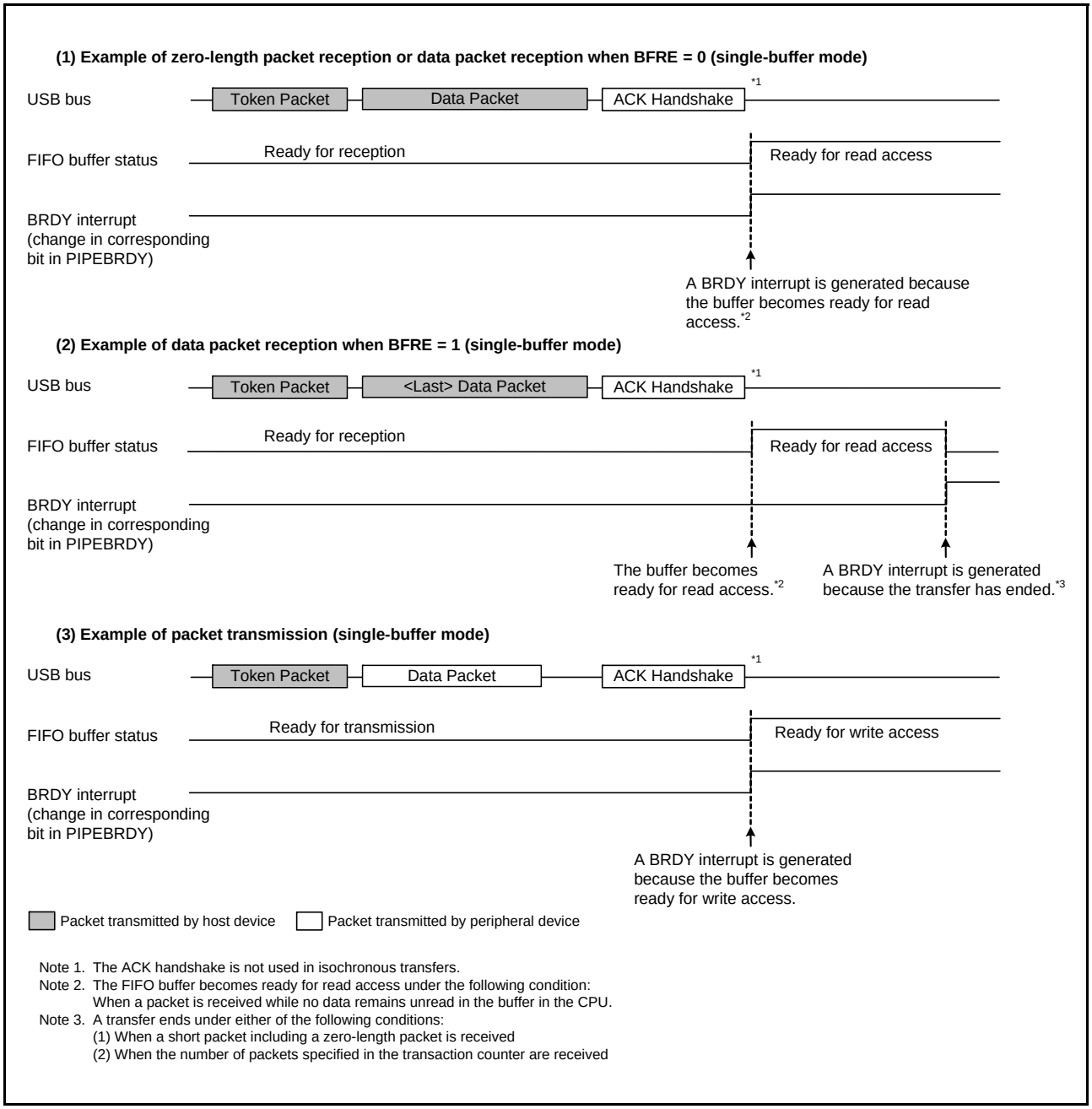


Figure 33.13 Timing of BRDY Interrupt Generation

The condition that USB clears the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting. Table 33.14 shows the condition for clearing the BRDY bit.

Table 33.14 Condition for Clearing BRDY Bit

BRDYM Bit	Condition for Clearing BRDY Bit
0	The USB clears the BRDY bit when software has cleared all bits in BRDYSTS.
1	The USB clears the BRDY bit when the BSTS bits for all piles have become 0.

33.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USB sets the corresponding PIPEnNRDY bit in NRDYSTS to 1. If the corresponding bit in NRDYENB has been set to 1 by software, the USB sets the INTSTS0.NRDY bit to 1 and generates a USB interrupt.
The following describes the conditions on which the USB generates the internal INTSTS0.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 function is selected. During setup transactions when the host controller function 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 function is selected.

(1) When Host Controller Function 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 bit and the FRMNUM.OVRN bit 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 bit to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 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 bit to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL (11b).

(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 bit corresponding to the pipe and the OVRN bit to 1.
  When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE bit 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 bit corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 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 bit 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 bit corresponding to the pipe and the CRCE bit to 1.
- When the STALL handshake is received.
  In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL.

(2) When Function Controller Function 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 bit 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 bit 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 bit to 1 and OVRN bit 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 bit to 1.
  However, during re-transmission (due to DATA-PID disagreement), 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 bit to 1.

Figure 33.14 shows the timing of NRDY interrupt generation when the function controller function is selected.

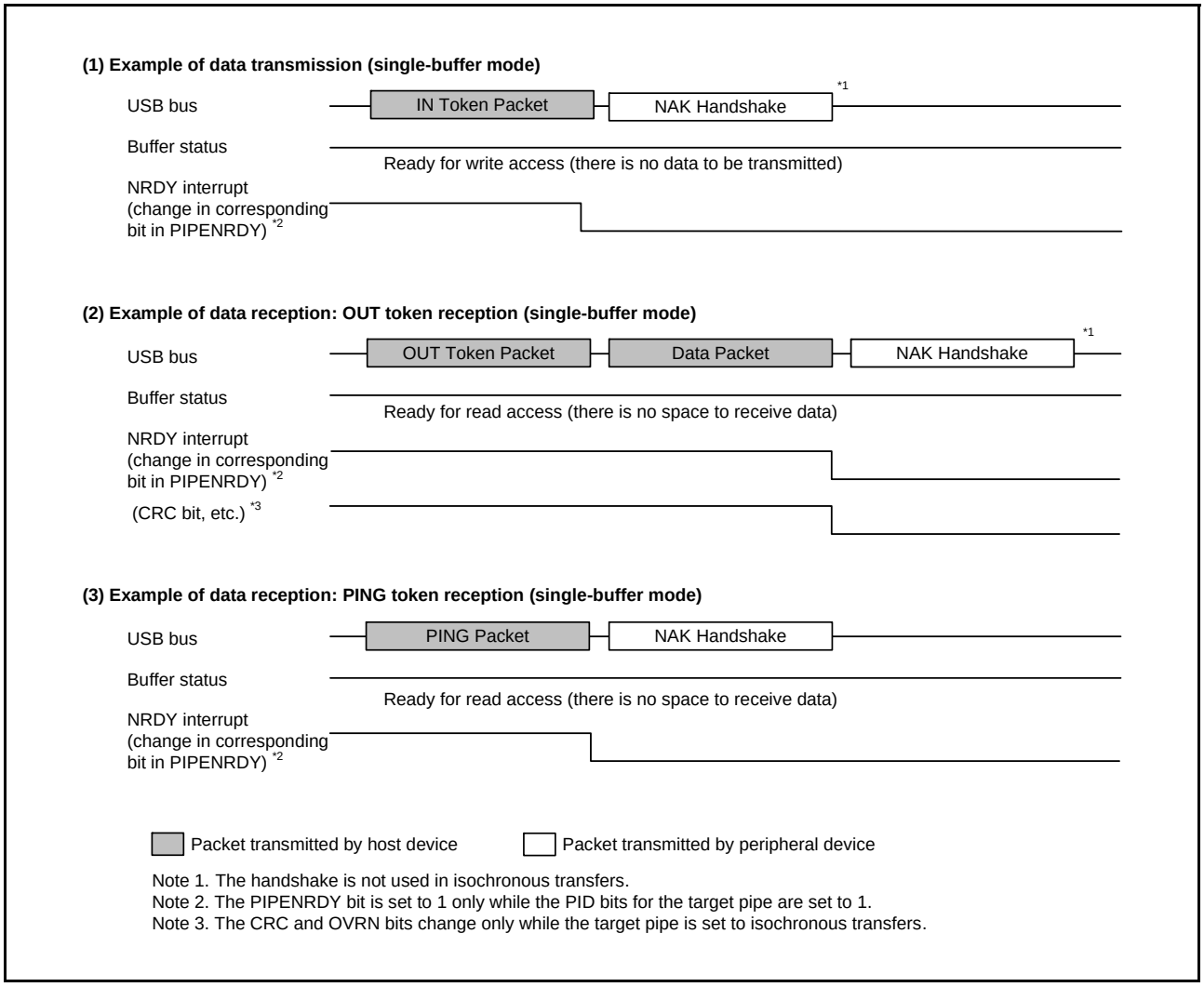


Figure 33.14 Timing of NRDY Interrupt Generation (When Function Controller Function is Selected)

33.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP bit to 1. If the corresponding bit in BEMPENB has been set to 1 by software, the USB sets the INTSTS0.BEMP bit 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 software (DMAC) 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 function 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 bit to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL (11). 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 bit clears the status.
  Writing 1 to the BEMPSTS.PIPEnBEMP bit has no effect.

Figure 33.15 shows the timing of BEMP interrupt generation when the function controller function is selected.

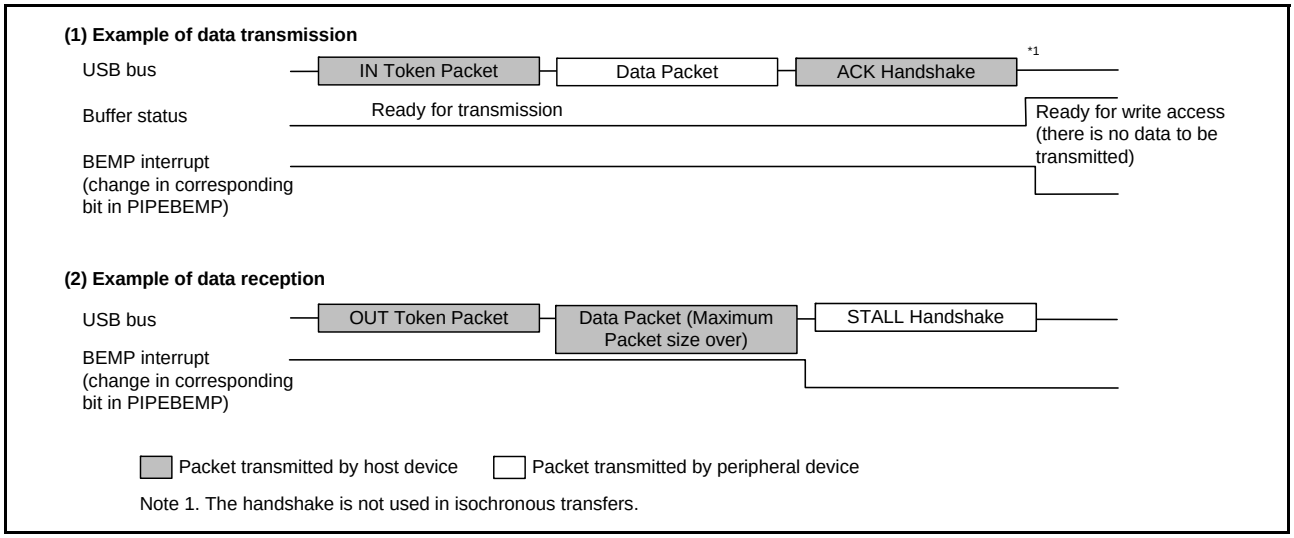
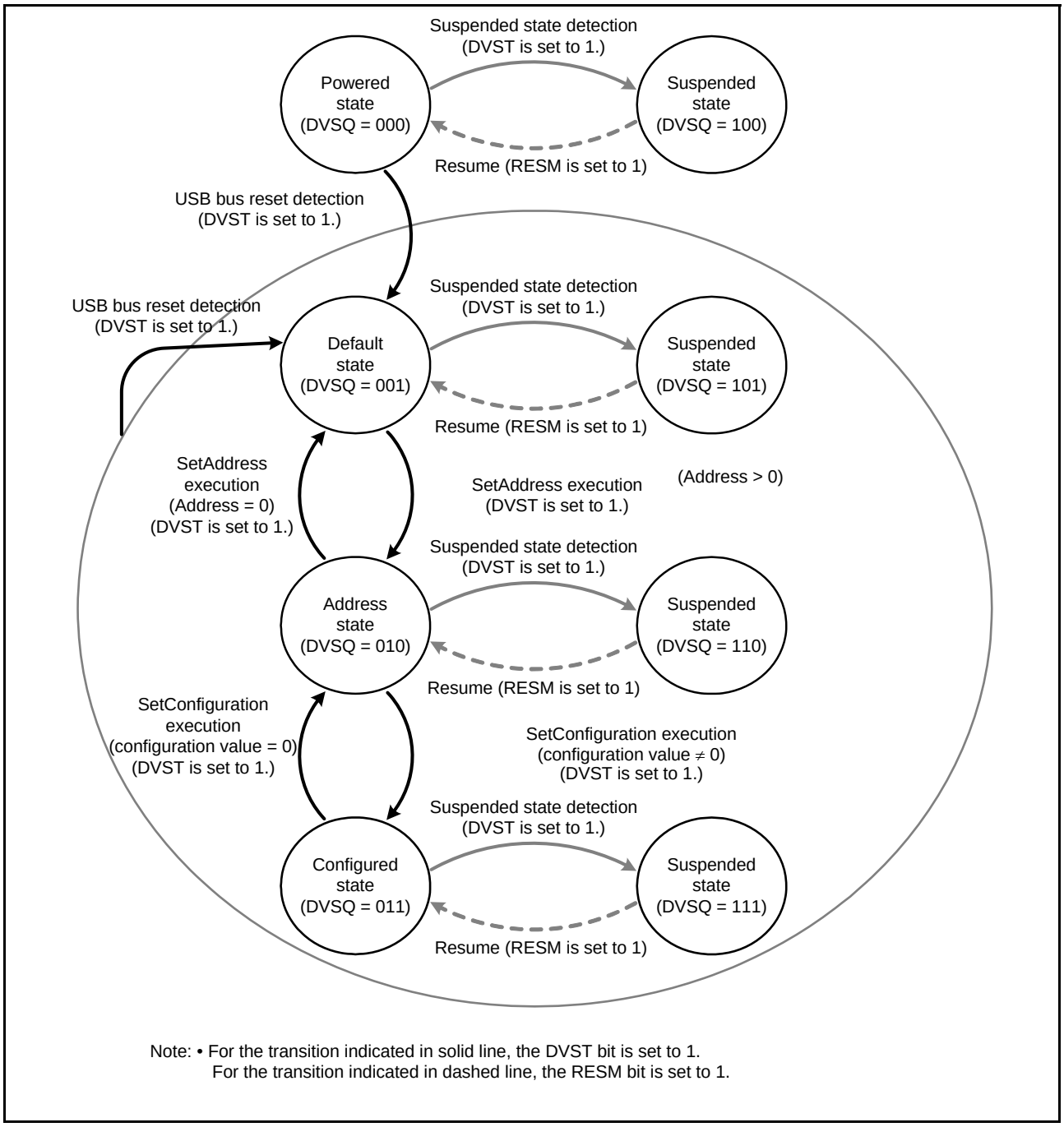


Figure 33.15 Timing of BEMP Interrupt Generation (When Function Controller Function is Selected)

33.3.3.4 Device State Transition Interrupt

Figure 33.16 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 DVSQ bits in INTSTS0.
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 function is selected. The device state transition interrupts can also be generated only when the function controller function is selected.



Note: • For the transition indicated in solid line, the DVST bit is set to 1.
For the transition indicated in dashed line, the RESM bit is set to 1.

Figure 33.16 Device State Transitions

33.3.3.5 Control Transfer Stage Transition Interrupt

Figure 33.17 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 CTSQ[2:0] bits in INTSTS0.

Control transfer stage transition interrupts are generated only when the function controller function 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 DATAPID = 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 DATAPID = 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 bit = 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.)

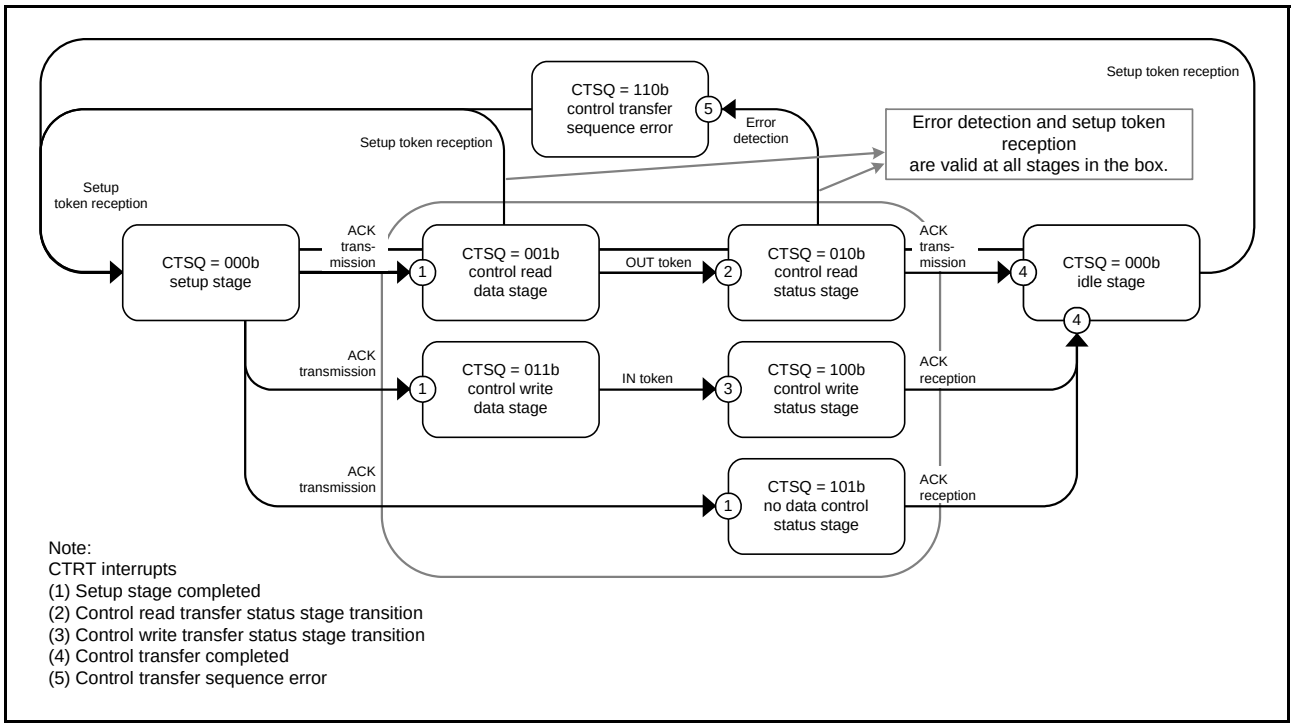


Figure 33.17 Control Transfer Stage Transitions

33.3.3.6 Frame Update Interrupt

With the host controller function selected, an interrupt is generated at the timing when the frame number is updated. With the function controller function selected, an SOFR interrupt is generated when the frame number is updated.
When the function controller function is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

33.3.3.7 VBUS Interrupt

When the USBm_VBUS pin level changes, a VBUS interrupt is generated. The level of the USBm_VBUS pin can be checked with the INTSTS0.VBSTS bit. 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 USBm_VBUS pin level.

33.3.3.8 Resume Interrupt

When the function controller function 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 function is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

33.3.3.9 OVRCR Interrupt

For port 0, 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.OCVMON[1:0] bits. The external power-supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For On-The-Go connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCR interrupt.

33.3.3.10 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 function is selected. The BCHG interrupt is generated regardless of whether the host controller function or function controller function is selected.

33.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller function 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). Software should terminate all pipes in which communications are currently carried out for the pertinent port 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.

33.3.3.12 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 function selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

33.3.3.13 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 function selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

33.3.3.14 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 s with the host controller function 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.

33.3.3.15 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). Software should terminate all pipes in which communications are currently carried out for the pertinent port 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 (n = 0 or 1).
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

33.3.4 Pipe Control

Table 33.15 lists the pipe setting items in the USB. With USB data transfer, data transfer has to be carried out using the logic pipe called 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 33.15 Pipe Setting Items

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE	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	Endpoint number	PIPE1 to PIPE9: Can be set A value other than 0000 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	Selects a device	Referenced only when the host controller function is selected.
	MXPS	Maximum packet size	Compliant with the USB standard.
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE9: Cannot be set
	IITV	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 function 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 PIPE3 to PIPE5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set Can be set only when the function controller function 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	Response PID	See section 33.3.4.6, 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	Transaction counter	PIPE1 to PIPE5: Can be set

33.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK).

Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State:

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN

In order to modify the above bits in the USB communication enabled (PID = 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] bit corresponding to the pipe to NAK.
3. Wait until the corresponding PBUSY bit is cleared 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 CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that Should Not be Set When CURPIPE[3:0] in FIFO-PORT is set:

- Bits in DCPCFG and DCPMAXP
- Bits in 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 in the port control register after the pipe information is modified.

33.3.4.2 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.

33.3.4.3 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 end point 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.

33.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the MXPS[8:0] in PIPEMAXP 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 the USB specification. 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 = 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.

33.3.4.5 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 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 = BUF, the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

33.3.4.6 Response PID

The PID[1:0] bits in DCPCTR and PIPEnCTR 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 function 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 function 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 function has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID = 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, see section 33.3.3.2, 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 = 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 function has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID = 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 = 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).

33.3.4.7 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 bit in DCPCTR and PIPEnCTR. 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 SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.
When the function controller function has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA0 is returned when the setup stage is ended and DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller function 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 function or function controller function is selected.

33.3.4.8 Response PID = NAK Function

The USB has a function that disables pipe operation (PID response = 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 (PID response = BUF).
The response PID = NAK function can be used only when bulk transfers are used.

33.3.4.9 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.

33.3.4.10 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 retuned 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.

33.3.4.11 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 bit 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 s) 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).

33.3.5 FIFO Buffer Memory

33.3.5.1 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 33.16 and Table 33.17 show the buffer status in the USB. The buffer memory status can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. The access 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 bit is valid for PIPE0 to PIPE5 in the transmitting direction.
When a transmitting pipe uses the double buffer configuration, software can read the BSTS bit to monitor the buffer memory status on the CPU side and the INBUFM bit 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 (DMAC) is slow, software can use the INBUFM bit to confirm the end of transmission.

Table 33.16 Buffer Status Indicated by the BSTS Bit

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 33.17 Buffer Status Indicated by the INBUFM Bit

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.

33.3.5.2 FIFO Buffer Clearing

Table 33.18 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.

Table 33.18 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.

(2) Buffer Memory Specifications (Single or Double Setting)

Either a single or double buffer configuration can be selected for PIPE1 to PIPE5, using the PIPECFG.DBLB bit.

33.3.5.3 FIFO Port Functions

Table 33.19 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] bits = 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] bits in the port control register.

Table 33.19 FIFO Port Function Settings

Register Name	Bit Name	Description
CFIFOSEL, DnFIFOSEL (n = 0 or 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 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 or 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 33.20 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 bit 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 33.20 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
	DMA 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 bit in the port control register = 1 should be checked after selecting a pipe.

33.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For PIPE1 to PIEP9, 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 33.21 shows the packet reception and buffer memory clearing processing by software for each of the various settings. As shown in Table 33.21, 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 33.21 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	Automatically cleared	Automatically cleared	Automatically cleared	Automatically cleared
Zero-length packet reception	Cleared by software	Cleared by software	Automatically cleared	Automatically cleared
Normal short packet reception	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared
Transaction count end	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared

33.3.6 Control Transfers (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.

33.3.6.1 Control Transfers when Host Controller Function is Selected

(1) Setup Stage

USQREQ, 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 cleared 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 DEVADDn corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0x2, make appropriate settings in DEVADD2; when PIPEMAXP.DEVSEL[3:0] = 0x5, make appropriate settings in DEVADD5.
When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN1 or SACK bit in INTSTS1), 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 SQMON bit in DCPCTR.

(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 bits = BUF in DCPCFG. 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] after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

33.3.6.2 Control Transfers when Function Controller Function is Selected

(1) Setup Stage

The USB always 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 bit to 1.
- Set the DCPCTR.PID[1:0] bits to 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 USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should always be carried out after setting the VALID bit = 0. In the VALID bit = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.
Using the function of the VALID bit, 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, see Figure 33.17.

(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 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
  The USB transmits a zero-length packet and receives an ACK response from the USB host.
- For control write transfers and no-data control transfers
  The USB receives a zero-length packet from the USB host and sends an ACK response.

(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.

- Any transfer other than a control read transfer: bmRequestType 00h
- Request error: wIndex 00h
- Any transfer other than a no-data control transfer: wLength 00h
- Request error: wValue > 7Fh
- Control transfer of a device state error: INTSTS0.DVSQ[2:0] = 011b (Configured)

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

33.3.7 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: see section 33.3.3.1, (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: see section 33.3.4.5, Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)
- Response PID = NAK function (PIPECFG.SHTNAK bit: see section 33.3.4.8, Response PID = NAK Function
- Auto response mode (PIPEnCTR.ATREPM bit: see section 33.3.4.9, Auto Response Mode)

33.3.8 Interrupt Transfers (PIPE6 to PIPE9)

When the function controller function is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.
When the host controller function is selected, the timing of issuing a token can be specified using the interval counter.

33.3.8.1 Interval Counter during Interrupt Transfers when Host Controller Function 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 USB controller initializes the interval counter under the following conditions.

- Power-on reset:
  The IITV[2:0] bits are initialized.
- Buffer memory initialization using the PIPEn.ACLRM bit:
  The IITV[2:0] bits are not initialized but the count value is initialized. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

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 UACT bit in DVSTCTR0 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 is set to NAK or 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.

33.3.9 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)

33.3.9.1 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, software can control them. Table 33.22 and Table 33.23 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 host controller function 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 function controller function 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 function 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 was received in frames other than the interval frame.

Table 33.22 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 function is selected and the function controller function is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN bit to 1 in both cases when host controller function is selected and function controller function is selected. When the function controller function 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 function is selected. It is not generated when the host controller function is selected.

Table 33.23 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 function is selected and the function controller function is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to STALL in both cases when the host controller function is selected and the function controller function is selected.

33.3.9.2 DATA-PID

When the function controller function 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

33.3.9.3 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 33.24 when the function controller function is selected. When the host controller function is selected, the USB generates the token issuance timing. When the host controller function is selected, the interval counter operation is the same as that in the interrupt transfer.

Table 33.24 Interval Counter Function when the Function Controller Function 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} frames.

(1) Counter Initialization when Function Controller Function is Selected

The USB initializes the interval counter under the following conditions.

- Power-on Reset
  The PIPEPERI.IITV[2:0] bits are initialized.
- Buffer memory initialization using the ACLRM bit
  The IITV[2:0] bits are not initialized but the count value is initialized. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in the IITV bits.

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 in the PID = BUF state.
2. An SOF is received after reception of data of an OUT token in the PID = BUF state.

Note that the interval counter is not initialized under the following conditions.

- When the PID[1:0] bits are set to NAK or 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 Host Controller Function 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} frames.
The USB starts counting the token issuance interval at the frame following the frame in which software has set the PID bits to BUF.

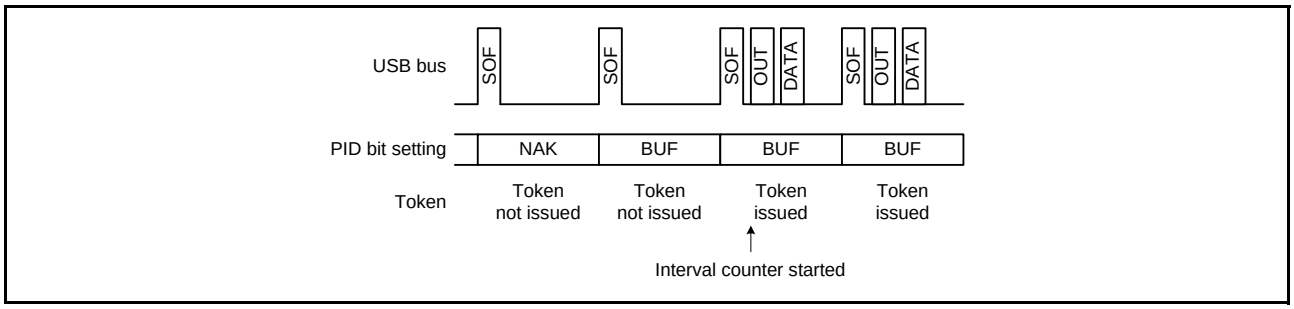


Figure 33.18 Token Issuance when IITV = 0

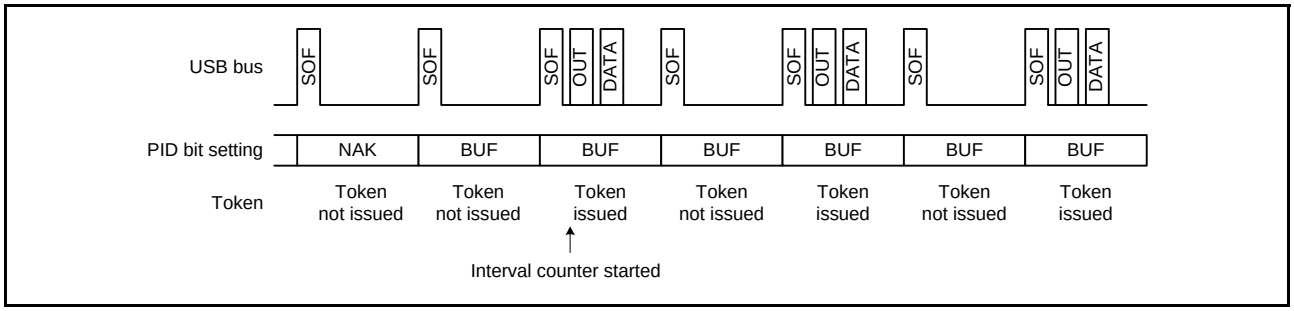


Figure 33.19 Token Issuance when IITV = 1

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 bit 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 software (DMAC) 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 bit 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 software (DMAC) 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 cleared to 0).
- When software sets the PIPEnCTR.ACLRM bit to 1

(3) Interval Counting and Transfer Control when Function Controller Function 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 bit is 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 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] bit as shown below.

- When the IITV = 0: The interval counting starts at the frame following the frame in which software has set the PID[1:0] bits for the selected pipe to BUF.

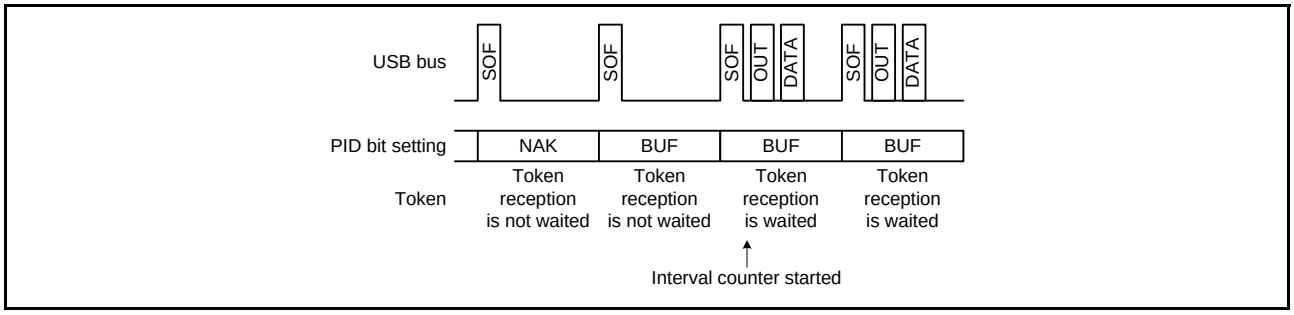


Figure 33.20 Relationship between Frames and Expected Token Reception when the IITV = 0

- When the IITV ≠ 0: 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 BUF.

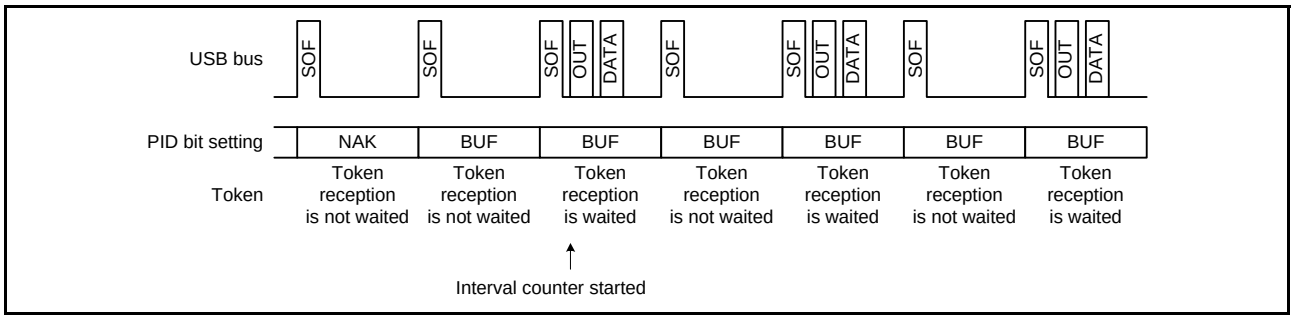


Figure 33.21 Relationship between Frames and Expected Token Reception when the IITV ≠ 0

(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 cleared to 000b).
- When software sets the PIPEnCTR.ACLRM bit to 1.
- When the USB detects a USB bus reset.

(4) Setup of Data to be Transmitted using Isochronous Transfer when Function Controller Function is Selected

With isochronous data transmission using the USB in function controller function, 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 33.22 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV = 0 (every frame) has been set.

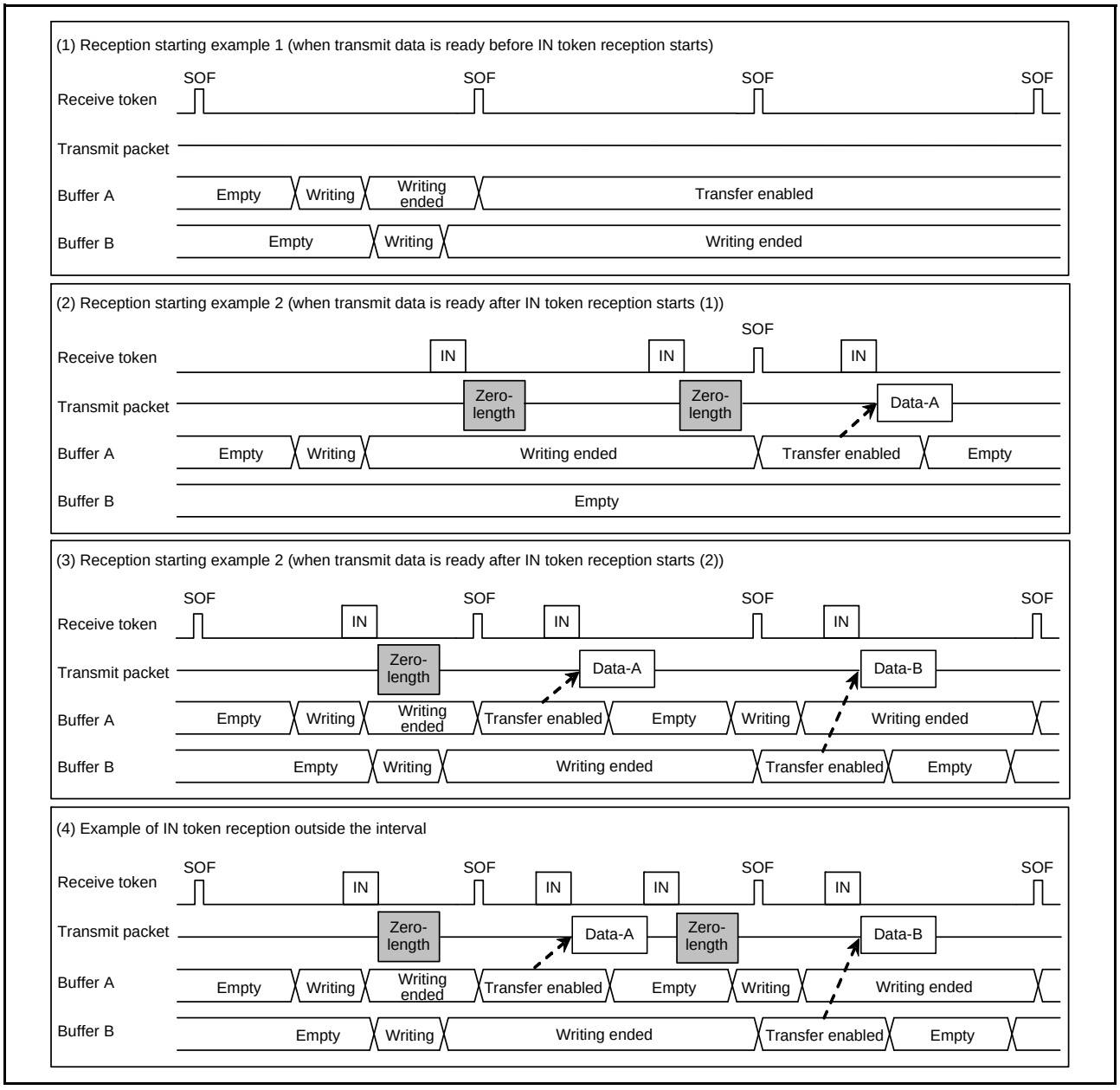


Figure 33.22 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush when Function Controller Function 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 = 0
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV ≠ 0
  The buffer flush operation is carried out after the first successful transaction.

Figure 33.23 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.

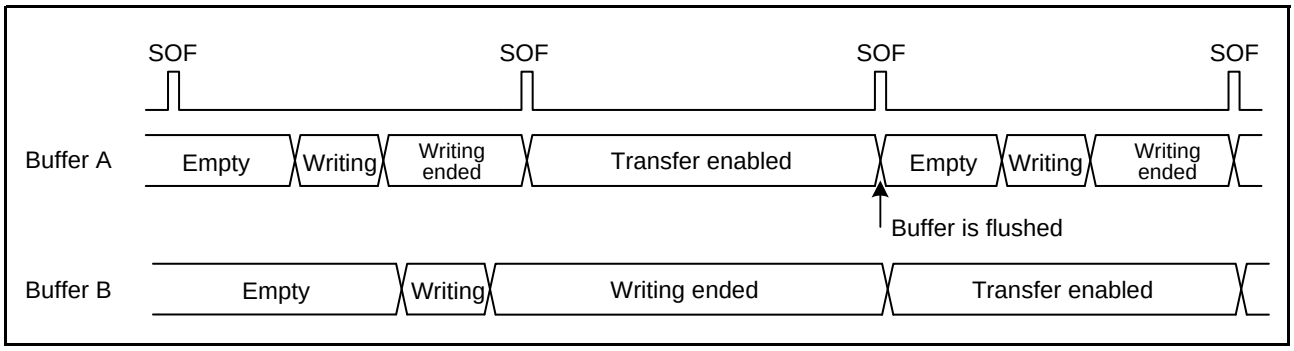


Figure 33.23 Example of Buffer Flush Operation

Figure 33.24 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 bit 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.

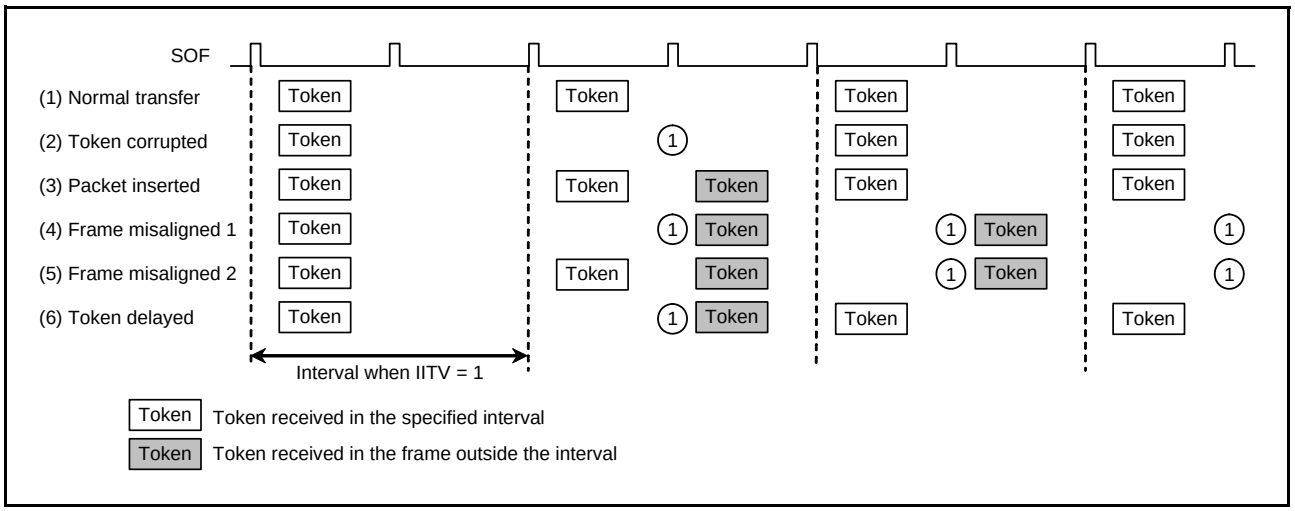


Figure 33.24 Example of Interval Error Occurrence when the IITV = 1

33.3.10 SOF Interpolation Function

When the function controller function 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 SYSCFG have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on 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 when full-speed operation is being used, the FRNM[10:0] bits FRMNUM0 is not updated.

33.3.11 Pipe Schedule

33.3.11.1 Conditions for Generating a Transaction

When the host controller function is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 33.25.

Table 33.25 Conditions for Generating a Transaction

	Conditions for Generation				
Transaction	**DIR**	**PID**	**IITV0**	**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.

33.3.11.2 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.

33.3.11.3 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.

33.4 Usage Notes

33.4.1 Setting the Module-Stop Function

Operation of the USB module can be prohibited or permitted by a bit in module-stop control register B (MSTPCRB). The setting after a reset is for operation of the USB module to be stopped. The registers are made accessible by release from the module-stop state. For details, see section 11., Low Power Consumption.

34. Serial Communications Interface (SCIc, SCId)

The RX63N/RX631 Group has 13 independent serial communications interface (SCI) channels. The SCI is configured as SCIc module (SCI0 to SCI11) and SCId module (SCI12).
The SCIc (SCI0 to SCI11) 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). Single-master operation as an simple I^2C bus interface and simple SPI interfaces are also supported.
As well as the functions of the SCIc module, the SCId module (SCI12) supports an extended serial protocol with a structure formed from Start Frames and Information Frames.

34.1 Overview

Table 34.1 lists the specifications of the SCIc module, Table 34.2 lists the specifications of the SCId module, and Table 34.3 lists the specifications of the individual SCI channels.
Figure 34.1 is a block diagram depicting SCI0 to SC11 (but not SCI5 and SCI6). Figure 34.2 is a block diagram depicting SCI5 and SCI6. Figure 34.3 is a block diagram depicting SCI12 (the SCId module).

Table 34.1 Specifications of SCIc (1/2)

Item		Specifications
Serial communications mode		• Asynchronous operation • Clock synchronous operation • Smart card interface • Simple I^2C bus • Simple SPI bus
Transfer speed		Bit rate specifiable with on-chip baud rate generator.
Full-duplex communications		Transmitter: Enables continuous transmission by double-buffering. Receiver: Enables continuous reception by double-buffering.
I/O pins		See Table 34.4 to Table 34.6.
Data transfer		Selectable as LSB-first or MSB-first transfer
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)
Power consumption reduction function		Module stop state can be set for each channel.
Asynchronous mode	Data length	7 or 8 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even, odd, or none
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.
	Break detection	Break can be detected by reading RXDn pin level directly in case of a framing error
	Clock source	Selectable from internal or external clock Enables transfer rate clock input from TMR (SCI5 and SCI6)
	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 errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.

Table 34.1 Specifications of SCIc (2/2)

Item		Specifications
Smart card interface mode	Error processing	An error signal can be automatically transmitted on detection of a parity error during reception
		Data can be automatically re-transmitted on 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	Up to 384 kbps
	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	Eight bits
	Detection of errors	Overrun errors
	SS input pin function*1	Applying the high level to the SS# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock sense are selectable.

Note 1. The SS input pin function in simple SPI mode is not available in products with the on-chip ROM capacity of 1.5 Mbytes or more, or with 176 pins or more.

Table 34.2 Specifications of SCId (1/2)

Item		Specifications
Serial communications mode		• Asynchronous operation • Clock synchronous operation • Smart card interface • Simple I^2C bus • Simple SPI bus
Transfer speed		Bit rate specifiable with on-chip baud rate generator.
Full-duplex communications		Transmitter: Enables continuous transmission by double-buffering. Receiver: Enables continuous reception by double-buffering.
Input/output pins		See Table 34.4 to Table 34.7.
Data transfer		Selectable as LSB-first or MSB-first transfer
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)
Power consumption reduction function		Module stop state can be set.
Asynchronous mode	Data length	7 or 8 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even, odd, or none
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.
	Break detection	Break can be detected by reading RXDn pin level directly in case of a framing error
	Clock source	Selectable from internal or external clock Enables transfer rate clock input from TMR
	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 errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.

Table 34.2 Specifications of SCId (2/2)

Item		Specifications
Smart card interface mode	Error processing	An error signal can be automatically transmitted on detection of a parity error during reception
		Data can be automatically re-transmitted on 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	Up to 384 kbps
	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 errors
	SS input pin function	Applying the high level to the SS# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock sense 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 RXDX12 • 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 SCIc when the extended serial mode control section is off.
	Timer function	• Usable as a reloading timer

Table 34.3 List of Functions of SCI Channels

Item	SCI0 to SCI4, SCI7 to SCI11	SCI5, SCI6	SCI12
Asynchronous mode	○	○	○
Clock synchronous mode	○	○	○
Smart card interface mode	○	○	○
Simple I^2C mode	○	○	○
Simple SPI bus	○	○	○
Extended serial mode	—	—	○
TMR clock input	—	○	○

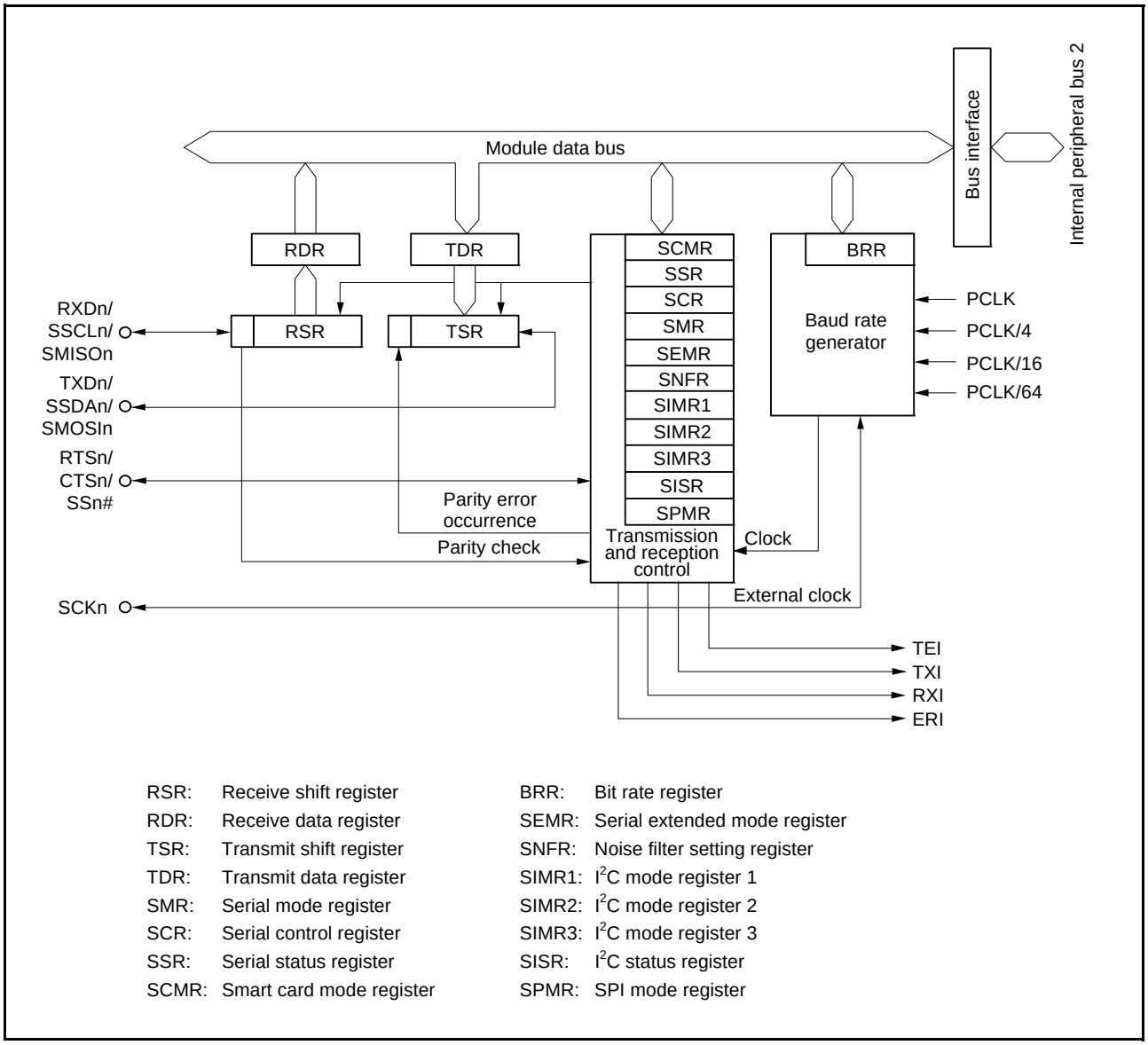
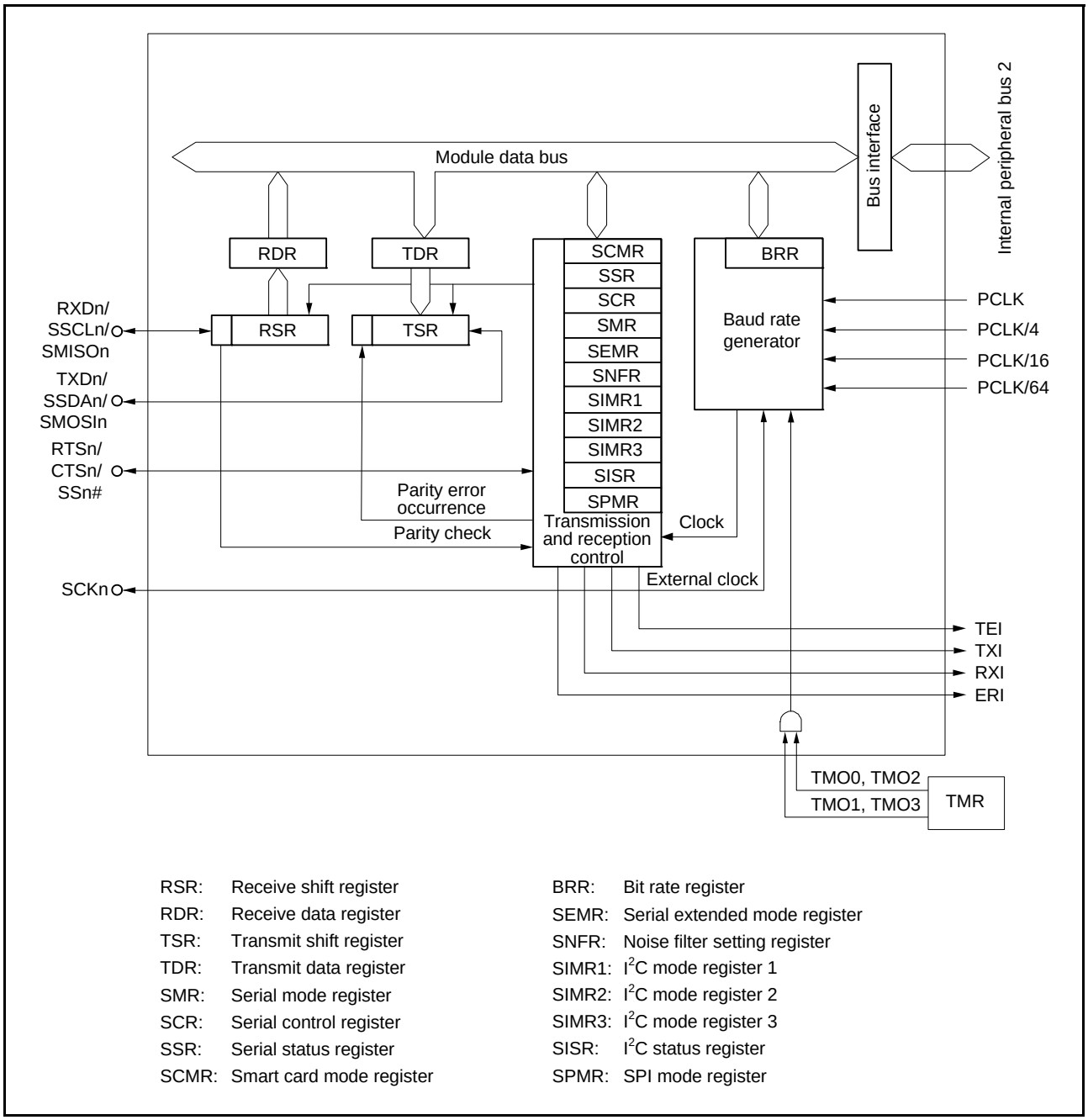


Figure 34.1 Block Diagram of SCI0 to SCI4, SCI7 to SCI11



RSR: Receive shift register
RDR: Receive data register
TSR: Transmit shift register
TDR: Transmit data register
SMR: Serial mode register
SCR: Serial control register
SSR: Serial status register
SCMR: Smart card mode register
BRR: Bit rate register
SEMR: Serial extended mode register
SNFR: Noise filter setting register
SIMR1: I^2C mode register 1
SIMR2: I^2C mode register 2
SIMR3: I^2C mode register 3
SISR: I^2C status register
SPMR: SPI mode register

Figure 34.2 Block Diagram of SCI5 and SCI6

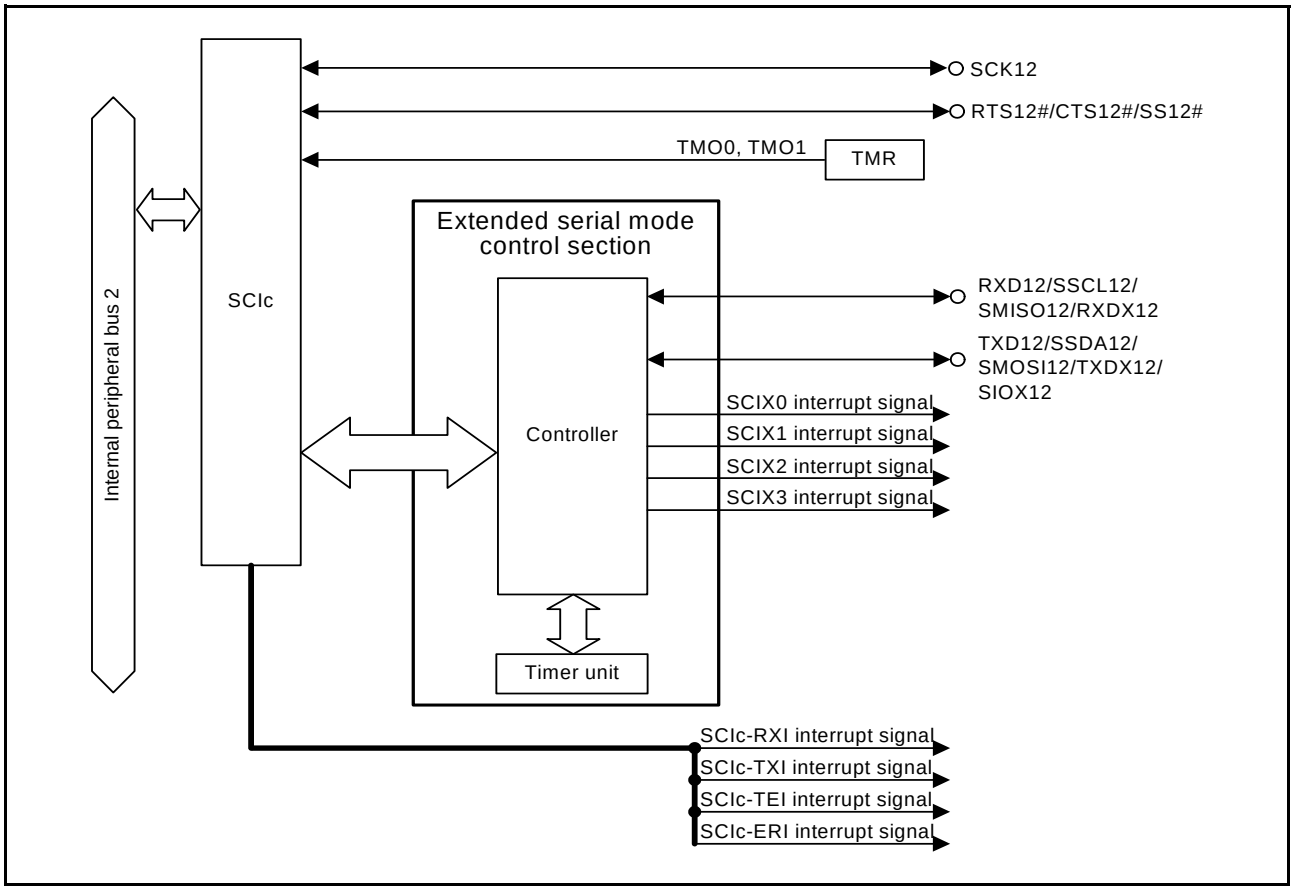


Figure 34.3 Block Diagram of SCI12 (SCId)

Table 34.4 to Table 34.7 list the pin configuration of the SCIs for the individual modes.

Table 34.4 Input and Output Pins of the SCIs (Asynchronous/Clock Synchronous Modes) (1/2)

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

Table 34.4 Input and Output Pins of the SCIs (Asynchronous/Clock Synchronous Modes) (2/2)

Channel	Pin Name	I/O	Function
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
SCI8	SCK8	I/O	SCI8 clock input/output
	RXD8	Input	SCI8 receive data input
	TXD8	Output	SCI8 transmit data output
	CTS8#/RTS8#	I/O	SCI8 transfer start control input/output
SCI9	SCK9	I/O	SCI9 clock input/output
	RXD9	Input	SCI9 receive data input
	TXD9	Output	SCI9 transmit data output
	CTS9#/RTS9#	I/O	SCI9 transfer start control input/output
SCI10	SCK10	I/O	SCI10 clock input/output
	RXD10	Input	SCI10 receive data input
	TXD10	Output	SCI10 transmit data output
	CTS10#/RTS10#	I/O	SCI10 transfer start control input/output
SCI11	SCK11	I/O	SCI11 clock input/output
	RXD11	Input	SCI11 receive data input
	TXD11	Output	SCI11 transmit data output
	CTS11#/RTS11#	I/O	SCI11 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 34.5 Pin Configuration of SCI (Simple I^2C Mode) (1/2)

Channel	Pin Name	I/O	Function
SCI0	SSCL0	I/O	SCI0 I^2C clock input/output
	SSDA0	I/O	SCI0 I^2C data input/output
SCI1	SSCL1	I/O	SCI1 I^2C clock input/output
	SSDA1	I/O	SCI1 I^2C data input/output
SCI2	SSCL2	I/O	SCI2 I^2C clock input/output
	SSDA2	I/O	SCI2 I^2C data input/output

Table 34.5 Pin Configuration of SCI (Simple I²C Mode) (2/2)

Channel	Pin Name	I/O	Function
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
SCI8	SSCL8	I/O	SCI8 I²C clock input/output
	SSDA8	I/O	SCI8 I²C data input/output
SCI9	SSCL9	I/O	SCI9 I²C clock input/output
	SSDA9	I/O	SCI9 I²C data input/output
SCI10	SSCL10	I/O	SCI10 I²C clock input/output
	SSDA10	I/O	SCI10 I²C data input/output
SCI11	SSCL11	I/O	SCI11 I²C clock input/output
	SSDA11	I/O	SCI11 I²C data input/output
SCI12	SSCL12	I/O	SCI12 I²C clock input/output
	SSDA12	I/O	SCI12 I²C data input/output

Table 34.6 Pin Configuration of SCI (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 34.6 Pin Configuration of SCI (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
SCI8	SCK8	I/O	SCI8 clock input/output
	SMISO8	I/O	SCI8 slave transmit data input/output
	SMOSI8	I/O	SCI8 master transmit data input/output
	SS8#	Input	SCI8 chip select input
SCI9	SCK9	I/O	SCI9 clock input/output
	SMISO9	I/O	SCI9 slave transmit data input/output
	SMOSI9	I/O	SCI9 master transmit data input/output
	SS9#	Input	SCI9 chip select input
SCI10	SCK10	I/O	SCI10 clock input/output
	SMISO10	I/O	SCI10 slave transmit data input/output
	SMOSI10	I/O	SCI10 master transmit data input/output
	SS10#	Input	SCI10 chip select input
SCI11	SCK11	I/O	SCI11 clock input/output
	SMISO11	I/O	SCI11 slave transmit data input/output
	SMOSI11	I/O	SCI11 master transmit data input/output
	SS11#	Input	SCI11 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 34.7 Pin Configuration of SCI (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

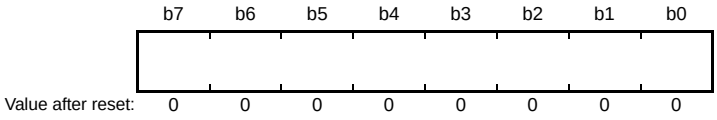
34.2 Register Descriptions

34.2.1 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 transferred to RDR automatically.
RSR cannot be directly accessed by the CPU.

34.2.2 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, SCI8.RDR 0008 A105h, SCI9.RDR 0008 A125h, SCI10.RDR 0008 A145h, SCI11.RDR 0008 A165h, SCI12.RDR 0008 B305h

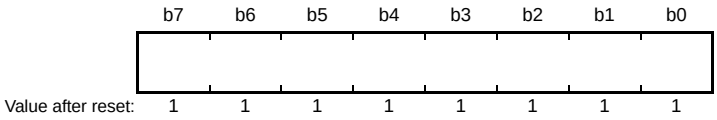


	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

RDR is an 8-bit register that stores receive data.
When the SCI has received one frame of serial data, it transfers the received serial data from RSR to RDR where it is stored. This allows RSR to 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.

34.2.3 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, SCI8.TDR 0008 A103h, SCI9.TDR 0008 A123h, SCI10.TDR 0008 A143h, SCI11.TDR 0008 A163h, 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 TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission.
The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission.
The CPU is able to read from or write to TDR at any time. Only write data for transmission to TDR once after each instance of the transmit data empty interrupt (TXI).

34.2.4 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.

34.2.5 Serial Mode Register (SMR)

Note: • Some bits in SMR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 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, SCI8.SMR 0008 A100h, SCI9.SMR 0008 A120h, SCI10.SMR 0008 A140h, SCI11.SMR 0008 A160h, 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 clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (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 in asynchronous mode) 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) 0: Selects 8 bits as the data length*2 1: Selects 7 bits as the data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode 1: Clock synchronous mode	R/W*4

Note 1. n is the decimal notation of the value of n in BRR (see section 34.2.9, Bit Rate Register (BRR)).
Note 2. In clock synchronous 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 TDR 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, see section 34.2.9, 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.
In clock synchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

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, SCI8.SMR 0008 A100h, SCI9.SMR 0008 A120h, SCI10.SMR 0008 A140h, SCI11.SMR 0008 A160h, SCI12.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 clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (n = 3)*1	R/W*3
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the BCP2 bit in SCMR. Setting values in BCP2 bit in SCMR and BCP[1:0] bits in SMR: BCP2 b3 b2 0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W*3
b4	PM	Parity Mode	(Valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity	R/W*3

Bit	Symbol	Bit Name	Description	R/W
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to data for transmission, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*3
b6	BLK	Block Transfer Mode	0: Normal mode operation 1: Block transfer mode operation	R/W*3
b7	GM	GSM Mode	0: Normal mode operation 1: GSM mode operation	R/W*3

Note 1. n is the decimal notation of the value of n in BRR (see section 34.2.9, Bit Rate Register (BRR)).
Note 2. S is the value of S in BRR (see section 34.2.9, Bit Rate Register (BRR)).
Note 3. 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, see section 34.2.9, 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 BCP2 bit in SCMR.
For details, see section 34.6.4, Receive Data Sampling Timing and Reception Margin.

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, see section 34.6.2, 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, see section 34.6.3, 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, see section 34.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 34.6.8, Clock Output Control.

34.2.6 Serial Control Register (SCR)

Note: • Some bits in SCR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 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, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, SCI10.SCR 0008 A142h, SCI11.SCR 0008 A162h, 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, SCI7 to SCI11 (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 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 or TMR clock • When an external clock is used, 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. (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. A 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 in TE and RE. While the SMR.CM 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 clearing the TEIE bit to 0.

In simple I^2C mode (when the IICM bit in SIMR is 1), 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 SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 34.4, Multi-Processor Communications Function.

When the receive data includes the MPB bit is SSR set to 0, 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 receive data includes the MPB bit set to 1, 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 RIE bit in SCR is set to 1), and setting the flags ORER and FER to 1 is enabled.

MPIE should be set 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 SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR 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 clearing the RIE bit to 0.

An ERI interrupt request can be cancelled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

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, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, SCI10.SCR 0008 A142h, SCI11.SCR 0008 A162h, 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	• When GM in SMR = 0 b1 b0 0 0: Output disabled (The SCKn pin is available for use as an I/O port in accord with the I/O port settings.) 0 1: Clock output 1 x: (Setting prohibited) • When GM in SMR = 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. A 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 in TE and RE. While the SMR.CM bit is 0, writing is enabled under any condition.

For details on interrupt requests, see section 34.11, 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, see section 34.6.8, 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 SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.
Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR 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 clearing the RIE bit to 0.
An ERI interrupt request can be cancelled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables notification of a TXI interrupt request.
Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

34.2.7 Serial Status Register (SSR)

Note: • Some bits in SSR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 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, SCI8.SSR 0008 A104h, SCI9.SSR 0008 A124h, SCI10.SSR 0008 A144h, SCI11.SSR 0008 A164h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	x	x	0	0	0	1	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame	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
b7, b6	—	Reserved	The read value is undefined. The write value should always be 1.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the RE bit in SCR is 0.

TEND Flag (Transmission End Flag)

Indicates completion of transmission.

[Setting conditions]

- Clearing of the SCR.TE bit to 0 (disabling serial transmission operations)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- The TDR is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When data for transmission are written to the TDR while the SCR.TE is 1
  When the TEND flag is cleared by writing the data for transmission to the TDR, read the TEND flag and check that it has actually been cleared to 0.

PER Bit (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 (after writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0 (which indicates that serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Bit (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 (after writing 0 to it, read the FER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the FER flag is not affected and retains its previous value.

ORER Bit (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 a 0 is written to ORER after reading ORER = 1 (after writing a 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

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, SCI8.SSR 0008 A104h, SCI9.SSR 0008 A124h, SCI10.SSR 0008 A144h, SCI11.SSR 0008 A164h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	x	x	0	0	0	1	0	0

x: Undefined

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
b7, b6	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

MPBT Bit (Multi-Processor Bit Transfer)

This bit should be set to 0 in smart card interface mode.

MPB Bit (Multi-Processor)

This bit is not used in smart card interface mode. It should be set to 0.

TEND Flag (Transmission 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 SCR.TE bit = 0 (disabling serial transmission operations)
  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 one 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 data for transmission are written to the TDR while the SCR.TE is 1

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 (After writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0 (which indicates that 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

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 (After writing 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

34.2.8 Smart Card Mode Register (SCMR)

Address(es): SCI0.SCMR 0008 A006h, SCI1.SCMR 0008 A026h, SCI2.SCMR 0008 A046h, SCI3.SCMR 0008 A066h, SCI4.SCMR 0008 A086h, SCI5.SCMR 0008 A0A6h, SCI6.SCMR 0008 A0C6h, SCI7.SCMR 0008 A0E6h, SCI8.SCMR 0008 A106h, SCI9.SCMR 0008 A126h, SCI10.SCMR 0008 A146h, SCI11.SCMR 0008 A166h, SCI12.SCMR 0008 B306h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	—	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: Serial communications interface mode 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is always read as 1. The write value should always 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*2	0: Transfer with LSB-first 1: Transfer with MSB-first	R/W*1
b6 to b4	—	Reserved	This bit is always read as 1. The write value should always 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. Setting values in the SCMR.BCP2 bit and SMR.BCP[1:0] bits BCP2 BCP1 BCP0 0 0 0: 93 clock cycles (S = 93)*3 0 0 1: 128 clock cycles (S = 128)*3 0 1 0: 186 clock cycles (S = 186)*3 0 1 1: 512 clock cycles (S = 512)*3 1 0 0: 32 clock cycles (S = 32)*3 (Initial Value) 1 0 1: 64 clock cycles (S = 64)*3 1 1 0: 372 clock cycles (S = 372)*3 1 1 1: 256 clock cycles (S = 256)*3	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. This bit is not valid in simple I2C mode.
Note 3. S is the value of S in BRR (see section 34.2.9, Bit Rate Register (BRR)).

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, asynchronous or clock synchronous 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 SMR.

SDIR Bit (Transmitted/Received Data Transfer Direction)

Selects the serial/parallel conversion format.

- Asynchronous mode
- Clock-synchronous mode
- Smart card interface mode
- Multiprocessor mode
- Simple SPI mode

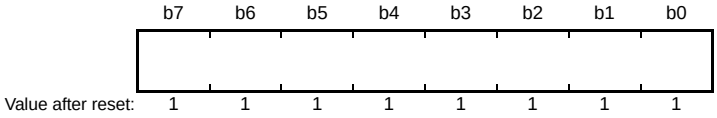
Set this bit to 1 if operation is to be in simple I2C mode.

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.

34.2.9 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, SCI8.BRR 0008 A101h, SCI9.BRR 0008 A121h, SCI10.BRR 0008 A141h, SCI11.BRR 0008 A161h, SCI12.BRR 0008 B301h



BRR 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 34.8 lists the relationships between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I^2C mode.

The initial value of BRR is FFh.

BRR can be read from by the CPU at all times, but it can be written to only when the TE and RE bits in SCR are 0.

Table 34.8 Relationships between N Setting in BRR and Bit Rate B

Mode	ABCS Bit in SEMR	BRR Setting	Error
Asynchronous, multi-processor transfer	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
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 I^2C*1		$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the SMR setting listed in the following table.
Note 1. 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.

Table 34.9 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I^2C	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 34.10 Clock Source Settings

SMR Setting		
CKS[1:0] Bits	Clock Source	n
0 0	PCLK clock	0
0 1	PCLK/4 clock	1
1 0	PCLK/16 clock	2
1 1	PCLK/64 clock	3

Table 34.11 Base Clock Settings in Smart Card Interface Mode

SCMR Setting	SMR Setting		
BCP2 Bit	BCP[1:0] Bits	Base Clock	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 34.12, Table 34.13 list sample N settings in BRR in normal asynchronous mode. Table 34.16 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 34.18. Examples of BRR (N) settings in smart card interface mode are listed in Table 34.20. Examples of BRR (N) settings in simple I^2C mode are listed in Table 34.22, Table 34.23. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 34.6.4, Receive Data Sampling Timing and Reception Margin. Table 34.17, Table 34.19 list the maximum bit rates with external clock input.

When the asynchronous mode base clock select bit (ABCS) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate is two times that of listed in Table 34.12, Table 34.15.

Table 34.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency PCLK (MHz)											
	8			9.8304			10			12		
Bit Rate (bps)	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
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34

Table 34.13 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	Operating Frequency PCLK (MHz)											
	12.288			14			16			17.2032		
Bit Rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	–0.17	3	70	0.03	3	75	0.48
150	2	159	0.00	2	181	0.16	2	207	0.16	2	223	0.00
300	2	79	0.00	2	90	0.16	2	103	0.16	2	111	0.00
600	1	159	0.00	1	181	0.16	1	207	0.16	1	223	0.00
1200	1	79	0.00	1	90	0.16	1	103	0.16	1	111	0.00
2400	0	159	0.00	0	181	0.16	0	207	0.16	0	223	0.00
4800	0	79	0.00	0	90	0.16	0	103	0.16	0	111	0.00
9600	0	39	0.00	0	45	–0.93	0	51	0.16	0	55	0.00
19200	0	19	0.00	0	22	–0.93	0	25	0.16	0	27	0.00
31250	0	11	2.40	0	13	0.00	0	15	0.00	0	16	1.20
38400	0	9	0.00	—	—	—	0	12	0.16	0	13	0.00

Note: • This is an example when the ABCS bit in SEMR is 0.
When the ABCS bit is set to 1, the bit rate is two times.

Table 34.14 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

	Operating Frequency PCLK (MHz)											
	17.2032			18			19.6608			20		
Bit Rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	75	0.48	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	16	1.20	0	17	0.00	0	19	-1.70	0	19	0.00
38400	0	13	0.00	0	14	–2.34	0	15	0.00	0	15	1.73

Table 34.15 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (4)

	Operating Frequency PCLK (MHz)											
	25			30			33			50		
Bit Rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	110	–0.02	3	132	0.13	3	145	0.33	3	221	–0.02
150	3	80	0.47	3	97	–0.35	3	106	0.39	3	162	–0.15
300	2	162	–0.15	2	194	0.16	2	214	–0.07	3	80	0.47
600	2	80	0.47	2	97	–0.35	2	106	0.39	2	162	–0.15
1200	1	162	–0.15	1	194	0.16	1	214	–0.07	2	80	0.47
2400	1	80	0.47	1	97	–0.35	1	106	0.39	1	162	–0.15
4800	0	162	–0.15	0	194	0.16	0	214	–0.07	1	80	0.47
9600	0	80	0.47	0	97	–0.35	0	106	0.39	1	40	–0.77
19200	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	80	0.47
31250	0	24	0.00	0	29	0	0	32	0	0	49	0.00
38400	0	19	1.73	0	23	1.73	0	26	–0.54	0	40	–0.77

Note: • This is an example when the ABCS bit in SEMR is 0.
When the ABCS bit is set to 1, the bit rate is two times.

Table 34.16 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0
30	937500	0	0
33	1031250	0	0
50	1562500	0	0

Note: • When the ABCS bit in SEMR is set to 1, the bit rate is two times.

Table 34.17 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)	
		when SEMR.ABCS bit = 0	when 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
50	12.500	781250	1562500

Note: • This is an example when the ABCS bit in SEMR is 0.

Table 34.18 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

	Operating Frequency PCLK (MHz)															
	8		10		16		20		25		30		33		50	
Bit Rate (bps)	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	194
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	3	77
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	155
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	2	77
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	124
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	61
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	124
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	49
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	24
1 M	0	0	—	—	0	3	0	4	—	—	—	—	—	—	—	—
2 M	0	0*1	—	—	0	1	—	—	—	—	—	—	—	—	—	—
2.5 M			0	0*1			0	1	—	—	0	2	—	—	0	4
4 M					0	0*1	—	—	—	—	—	—	—	—	—	—
5 M							0	0*1	—	—	—	—	—	—	—	—
6.25 M									0	0*1	—	—	—	—	0	1
7.5 M											0	0*1	—	—	—	—
8.25 M													0	0*1	—	—

Space: Setting prohibited.
— : Can be set, but there will be error.
Note 1. Continuous transmission or reception is not possible.

Table 34.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
25	4.1667	4166666.7
30	5.0000	5000000.0
33	5.5000	5500000.0
50	8.3333	8333333.3

Table 34.20 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
	50.00	0	27	0.00

Table 34.21 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0
25.00	33602	0	0
30.00	40323	0	0
33.00	44355	0	0
50.00	67205	0	0

Table 34.22 BRR Settings for Various Bit Rates (Simple I²C Mode)(1)

	Operating Frequency PCLK (MHz)											
	8			10			16			20		
Bit Rate (bps)	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
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7
350 k										0	1	–10.7

Table 34.23 BRR Settings for Various Bit Rates (Simple I²C Mode)(2)

	Operating Frequency PCLK (MHz)											
	25			30			33			50		
Bit Rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	19	–2.3	1	23	–2.3	1	25	–0.8	2	9	–2.3
25 k	1	7	–2.3	1	9	–6.3	1	10	–6.3	2	3	–2.3
50 k	1	3	–2.3	1	4	–6.3	1	5	–14.1	2	1	–2.3
100 k	1	1	–2.3	1	2	–21.9	1	2	–14.1	1	3	–2.3
250 k	0	3	–21.9	0	3	–6.3	0	4	–17.5	0	6	–10.7
350 k	0	2	–25.6	0	2	–10.7	0	2	–1.8	0	4	–10.7

Table 34.24 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)(1)

	Operating Frequency PCLK (MHz)											
	8			10			16			20		
Min. Widths at High/Low Level for SCL (µs)	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.5/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.2/20.80	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

Table 34.25 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)(2)

	Operating Frequency PCLK (MHz)											
	25			30			33			50		
Min. Widths at High/Low Level for SCL (µs)	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width
10 k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	2	9	44.80/51.20
25 k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	2	3	17.92/20.48
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	2	1	8.96/10.24
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	1	3	4.48/5.12
250 k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	6	1.96/2.24
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	4	1.40/1.60

34.2.10 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, SCI8.SEMR 0008 A107h, SCI9.SEMR 0008 A127h, SCI10.SEMR 0008 A147h, SCI11.SEMR 0008 A167h, SCI12.SEMR 0008 B307h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFEN	ABCS	—	—	—	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: TMR clock input (valid only for SCI5, SCI6, and SCI12) The following table lists the correspondence between SCI channels and compare match outputs.	R/W*1
b3 to b1	—	Reserved	These bits are 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
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SCI	TMR	Compare Match Output
SCI5	Unit 0	TMO0, TMO1
SCI6	Unit 1	TMO2, TMO3
SCI12	Unit 0	TMO0, TMO1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

SEMR selects the clock source for 1-bit period in 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.
Figure 34.4 shows a setting example when the TMOn output of TMRn (n = 0 to 3) is selected.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.
The ACS0 bit is valid in asynchronous mode (CM bit in SMR = 0) and when an external clock input is selected (CKE[1:0] bits in SCR = 10b or 11b). An external clock input or internal TMR clock input can be selected.
Clear the ACS0 bit to 0 in other than asynchronous mode.
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 always be 0.

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.

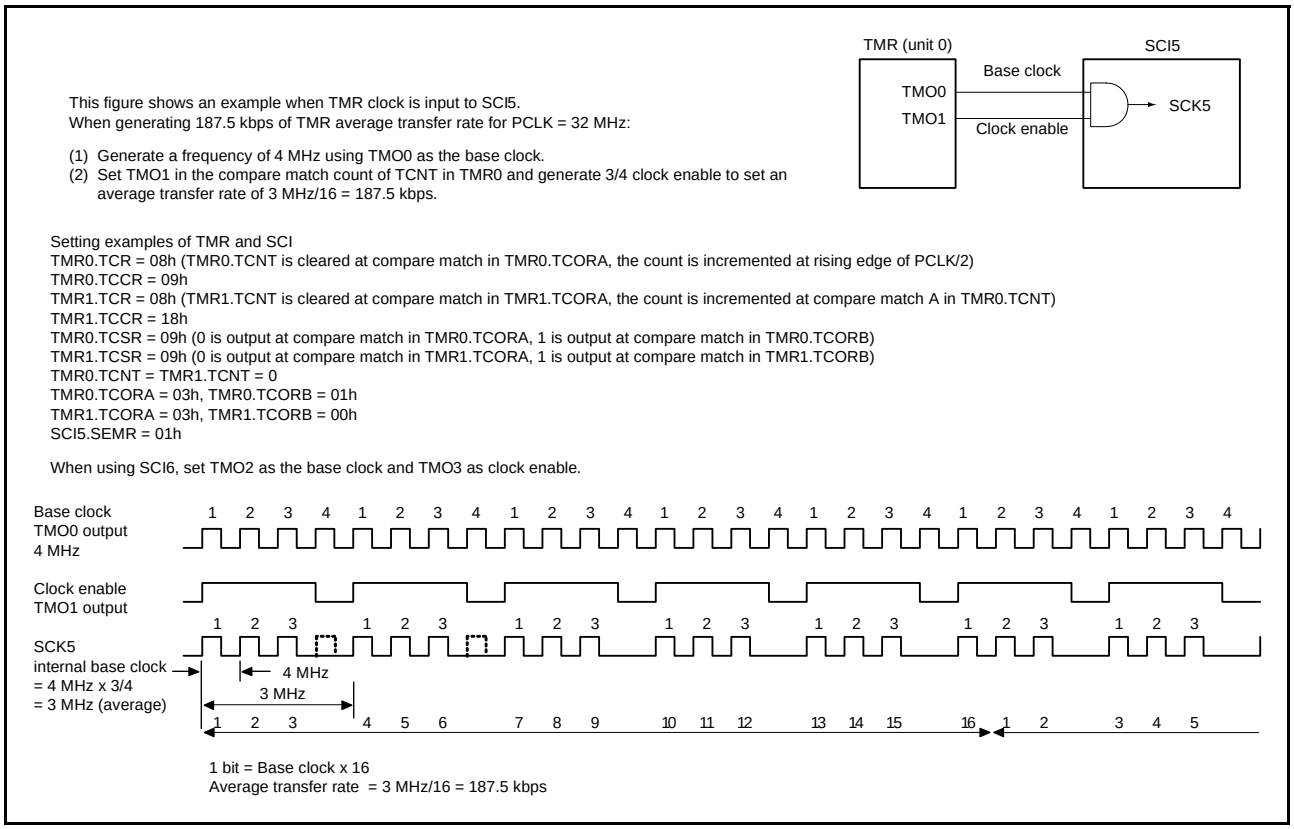
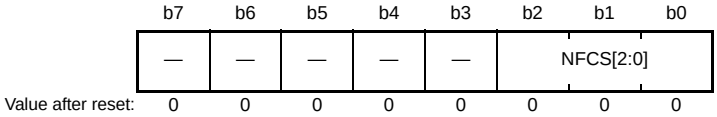


Figure 34.4 Example of Average Transfer Rate Setting when TMR Clock is Input

34.2.11 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, SCI8.SNFR 0008 A108h, SCI9.SNFR 0008 A128h, SCI10.SNFR 0008 A148h, SCI11.SNFR 0008 A168h, SCI12.SNFR 0008 B308h



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 one is used with the noise filter. In simple I²C mode, the standard settings for the clock source selected for the on-chip baud rate generator are given below. b2 b0 0 0 1: The clock signal divided by one is used with the noise filter. 0 1 0: The clock signal divided by two is used with the noise filter. 0 1 1: The clock signal divided by four is used with the noise filter. 1 0 0: The clock signal divided by eight is used with the noise filter. Other values: Do not make settings other than those listed above.	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 (disabling reception and transmission).

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.

34.2.12 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, SCI8.SIMR1 0008 A109h, SCI9.SIMR1 0008 A129h, SCI10.SIMR1 0008 A149h, SCI11.SIMR1 0008 A169h, SCI12.SIMR1 0008 B309h

Bit	Symbol	Bit Name	Description	R/W
b0	IICM	Simple I²C Mode Select	SMIF IICM 0 0: Serial interface mode (in asynchronous, 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 Delay Output 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 (disabling reception and transmission).

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 SCMR, 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 SMR.CKS[1:0] 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.

34.2.13 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, SCI8.SIMR2 0008 A10Ah, SCI9.SIMR2 0008 A12Ah, SCI10.SIMR2 0008 A14Ah, SCI11.SIMR2 0008 A16Ah, 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 (disabling serial reception and transmission).

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 bit 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.

34.2.14 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, SCI8.SIMR3 0008 A10Bh, SCI9.SIMR3 0008 A12Bh, SCI10.SIMR3 0008 A14Bh, SCI11.SIMR3 0008 A16Bh, 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	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2,*3,*4	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2,*3,*4	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: All request generation has been completed.	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 SSDA pin. 1 1: Place the SSDA 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 SSCL pin. 1 1: Place the SSCL 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.

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 execution for the generation of a condition, this bit indicates the state of generation being completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after clearing the IICSTIF flag to 0.

If the TEIE bit in the SCR is enabling interrupt requests, an STI is output on completion of generation of the start, restart, or stop condition when the IICSTIF flag is 1.

[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 zero 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 IICM bit in SIMR1 (when operation is not in simple I^2C mode)
- Writing 0 to the TE bit in SCR

IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.

Set the IISCDAS and IICSCLS 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 and IISCDAS bits to the same value during normal operations.

34.2.15 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, SCI8.SISR 0008 A10Ch, SCI9.SISR 0008 A12Ch, SCI10.SISR 0008 A14Ch, SCI11.SISR 0008 A16Ch, 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 IICACK flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

34.2.16 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, SCI8.SPMR 0008 A10Dh, SCI9.SPMR 0008 A12Dh, SCI10.SPMR 0008 A14Dh, SCI11.SPMR 0008 A16Dh, 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	SS Pin Function Enable	0: SS pin function is disabled. 1: SS pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS pin function is disabled (RTS output function is enabled). 1: CTS pin 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 always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Clock polarity is not inverted. 0: Clock polarity is inverted.	R/W*1
b7	CKPH	Clock Phase Select	0: Clock is not delayed. 0: 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 (disabling reception and transmission).
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 (SS 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 SS# 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 CTS bits to “enabled” (even if this setting is made, the functions will be disabled).

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 CTS and SSE bits to “enabled” (even if this setting is made, the functions will be disabled).

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 are 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 SS# 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. See Figure 34.52 for details.
Clear 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. See Figure 34.52 for details.
Clear the bit to 0 in other than simple SPI mode and clock synchronous mode.

34.2.17 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 enters the following states:

- the extended serial mode control section is initialized

Table 34.26 Settings of the ESME Bits and Guaranteed Operation by Timer Operation Mode

ESME bit	Timer Mode	Break Field Low Width Judgment Mode	Break Field Low Width Output Mode
0	○*1	×	×
1	○	○	○

○: Guarantee of operation is necessary. ×: Guarantee of operation is not necessary.
Note 1. Operation is only possible with PCLK selected.

34.2.18 Control Register 0 (CR0)

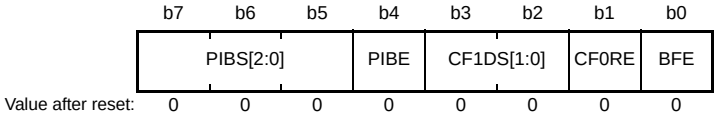
Address(es): SCI12.CR0 0008 B321h

	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

34.2.19 Control Register 1 (CR1)

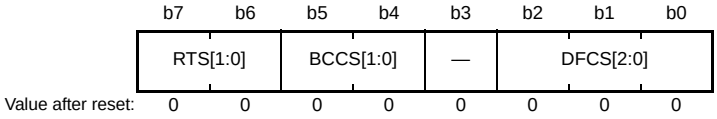
Address(es): SCI12.CR1 0008 B322h



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

34.2.20 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B320h



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 is enabled (SCI base clock). 0 1 0: Filter is enabled (SCI base clock/8). 0 1 1: Filter is enabled (SCI base clock/16). 1 0 0: Filter is enabled (SCI base clock/32). 1 0 1: Filter is enabled (SCI base clock/64). 1 1 0: Filter is enabled (SCI base clock/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	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	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 8th cycle of SCI base clock	R/W

Note 1. The period of the SCI base clock is 1/16 of a single data period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single data period when the SCI12.SEMR.ABCS is 1. To use the SCI base clock, set the SCI12.SCR.TE bit to 1.

34.2.21 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 Bits (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

34.2.22 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 Selection)

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.

34.2.23 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

34.2.24 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 BCDF 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

34.2.25 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

34.2.26 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

CF0DR is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

34.2.27 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

34.2.28 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 read-only register that holds the value received in Control Field 0. Writing to this register from the CPU or DTC is not possible.

34.2.29 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.

34.2.30 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.

34.2.31 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
b6	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

34.2.32 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 read-only register that holds the value received in Control Field 1. Writing to this register from the CPU or DTC is not possible.

34.2.33 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

34.2.34 Timer Mode Register (TMR)

Address(es): SCI12.TMR 0008 B31F331h

	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: Writing is to the reload register and the counter. 1: Writing is only to the reload register	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	These bits are read as 0. The write value should be 0.	R/W

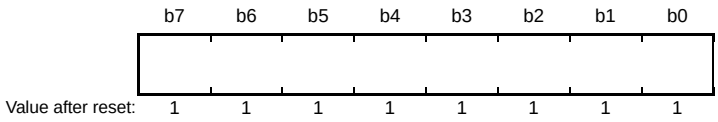
Note 1. Stop the counter (TCST = 0) before writing to the TOMS[1:0] or TCSS[2:0] bits.

TWRC Bit (Counter Write Control)

These bits determine whether a value written to TPRE or TCNT is only written to the reload register or is written to both the reload register and the counter.

34.2.35 Timer Prescaler Register (TPRE)

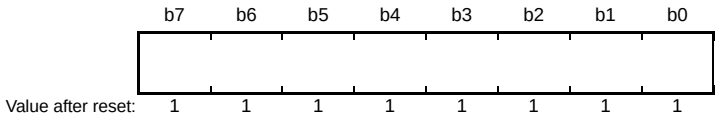
Address(es): SCI12.TPRE 0008 B332h



TPRE consists of an eight-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, so in writing, transfer is to the reload register and in reading, transfer is of the counter value from the read buffer.
It takes 1 system operating clock cycle to load a value from the reload register to the counter.

34.2.36 Timer Count Register (TCNT)

Address(es): SCI12.TCNT0008 B333h



TCNT consists of an eight-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, so in writing, transfer is to the reload register and in reading, transfer is of the counter value from the read buffer.
It takes 1 system operating clock cycle to load a value from the reload register to the counter.

34.3 Operation in Asynchronous Mode

Figure 34.5 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, and when it goes to the space state (low level), recognizes a start bit and starts serial communications.
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.

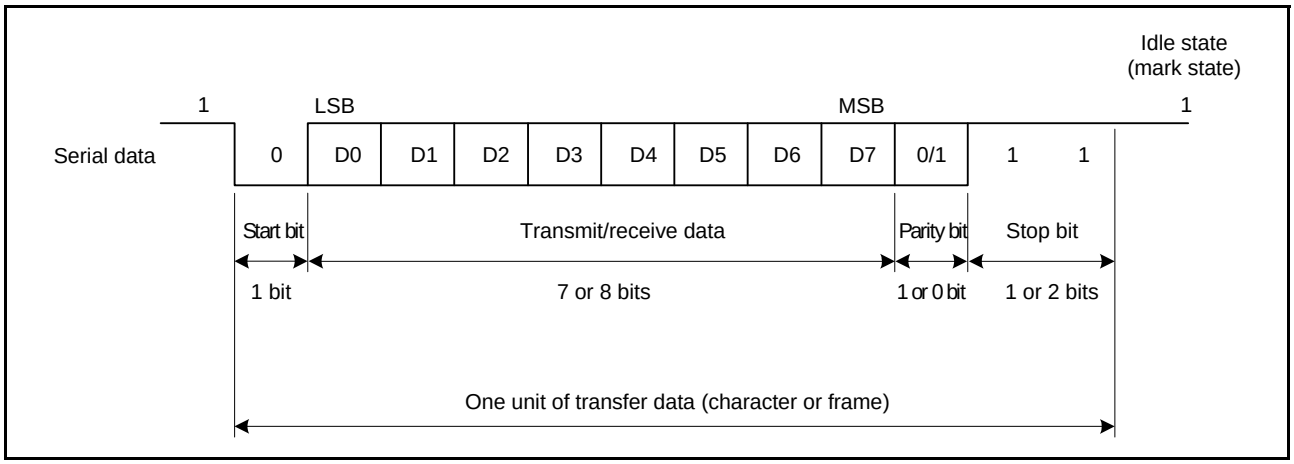


Figure 34.5 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, Two Stop Bits)

34.3.1 Serial Data Transfer Format

Table 34.27 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 12 transfer formats can be selected according to the SMR setting. For details of multi-processor function, see section 34.4, Multi-Processor Communications Function.

Table 34.27 Serial Transfer Formats (Asynchronous Mode)

SMR Setting				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STOP
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

34.3.2 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 34.6. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{|D - 0.5|}{N}\ (1+F) \right| \times 100\ [\%] \quad \cdots \text{ Formula (1)}$$

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 12)
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.

$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\%$

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 SEMR 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

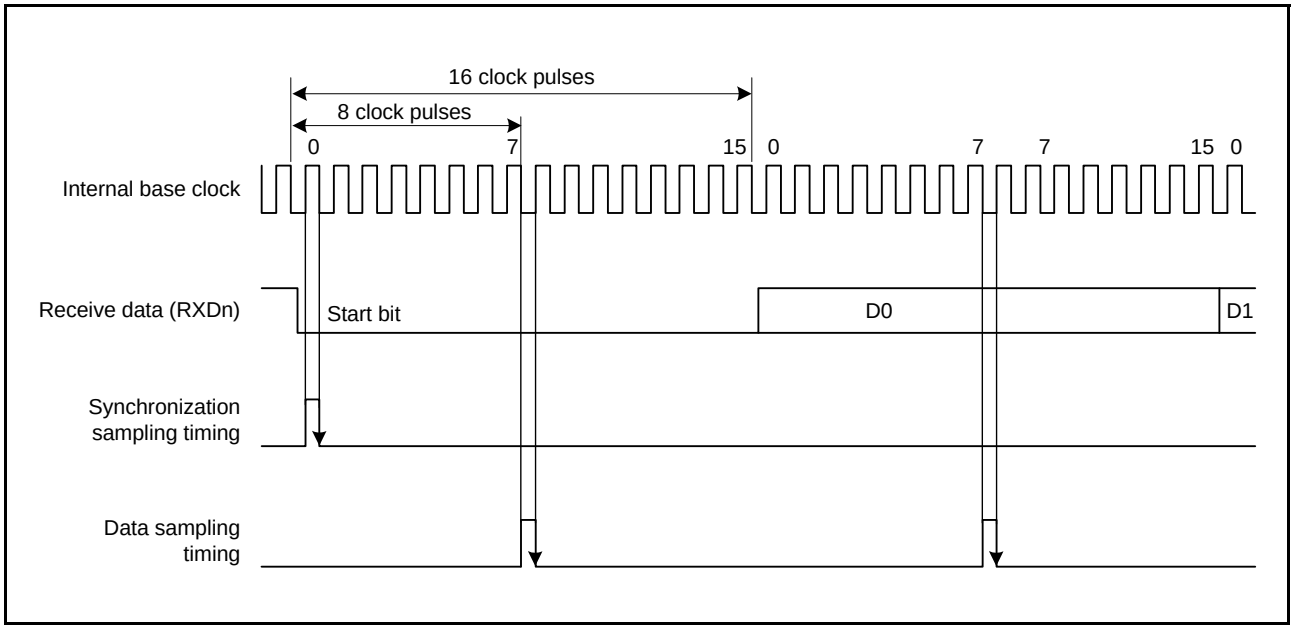


Figure 34.6 Receive Data Sampling Timing in Asynchronous Mode

34.3.3 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 SMR and the CKE[1:0] bits in SCR.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when ABCS in SEMR = 0) and 8 times the bit rate (when ABCS in SEMR = 1). In addition, when an external clock is specified, the base clock of TMR0 and TMR1 can be selected by the ACS0 bit in SEMR of SCIn (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 34.7.

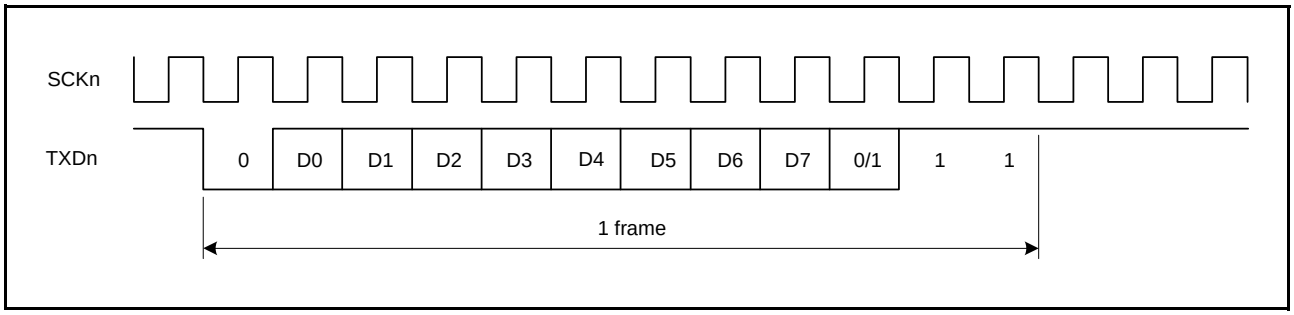


Figure 34.7 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

34.3.4 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 low 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]
Satisfaction of all conditions listed below

- The value of the RE bit in the SCR 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]

- Any of the conditions for the low level not being satisfied

34.3.5 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value “00h” to SCR and then continue through the procedure for SCI given in the sample flowchart (Figure 34.8). Whenever the operating mode or transfer format is changed, SCR 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 clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.
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.

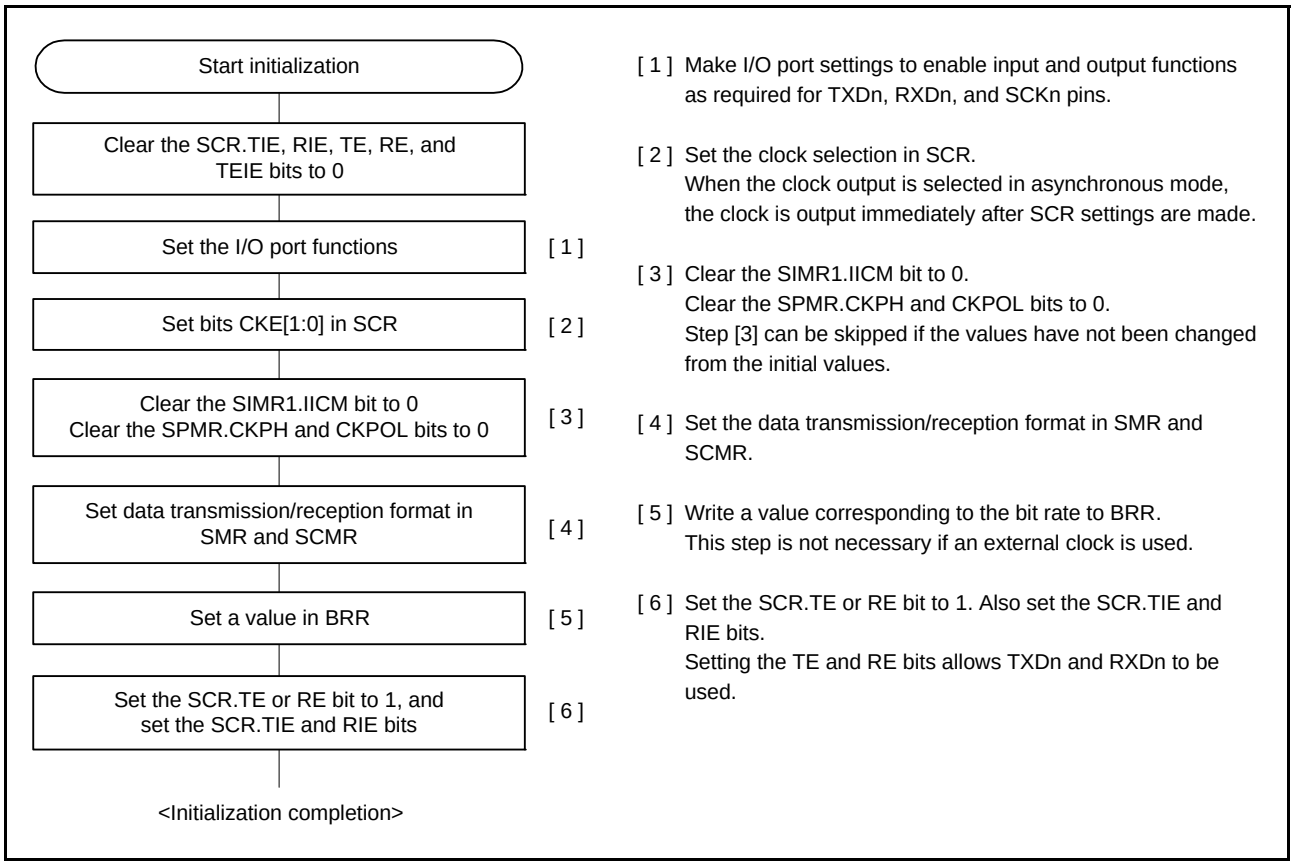


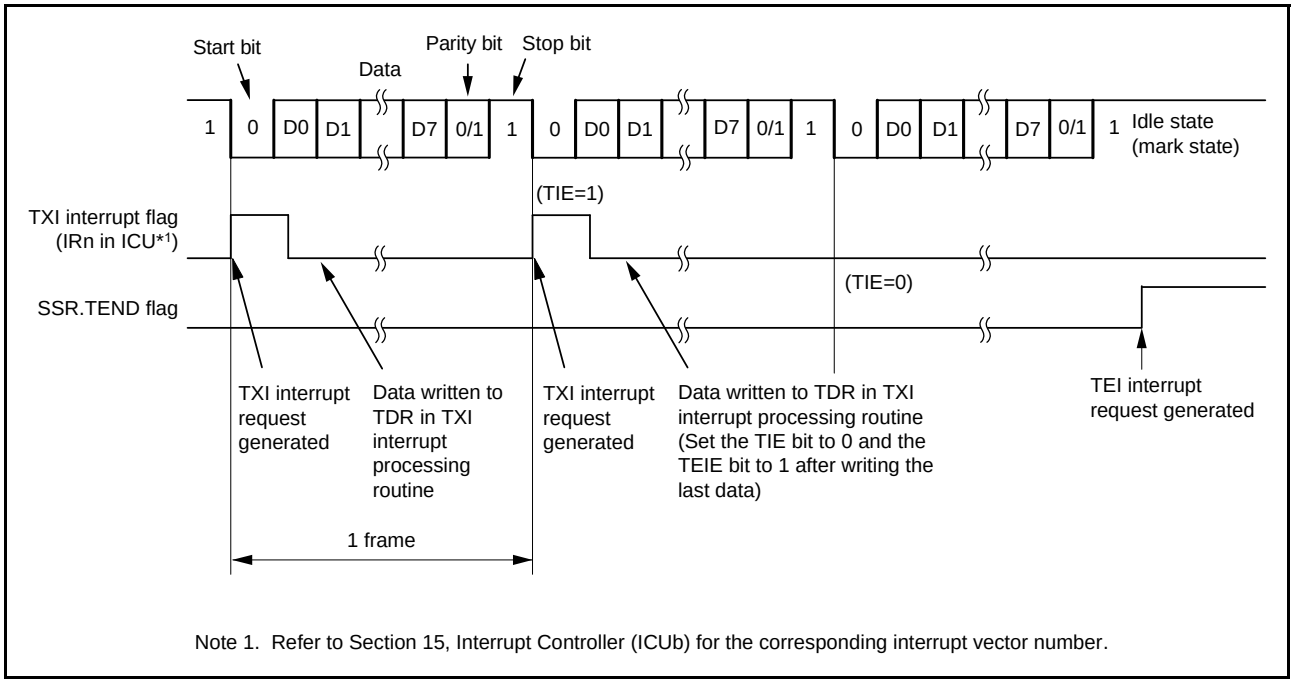
Figure 34.8 Sample SCI Initialization Flowchart (Asynchronous Mode)

34.3.6 Serial Data Transmission (Asynchronous Mode)

Figure 34.9 shows 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 TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SPMR is set to 0 (disabling the CTS function) and a low level on the CTS# pin causes data transfer from TDR to TSR. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next data for transmission to TDR in the TXI interrupt processing routine before transmission of the current data for transmission is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (disabling TXI requests) and the SCR.TEIE bit to 1 (enabling TEI requests) after the last of the data to be transmitted are written to the TDR from the processing 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) TDR at the time of stop bit output.
5. When TDR is updated, setting of the CTSE bit in SPMR to 0 (CTS function disabled) or a low level input on the CTSn# pin cause the next transfer of the next data for transmission from TDR to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the TEIE flag in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.

Figure 34.10 shows a sample flowchart for serial transmission in asynchronous mode.



Note 1. Refer to Section 15, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 34.9 Example of Operation for Serial Transmission in Asynchronous Mode (from the Middle of Transmission until Transmission Completion) (Example with 8-Bit Data, Parity, One Stop Bit)

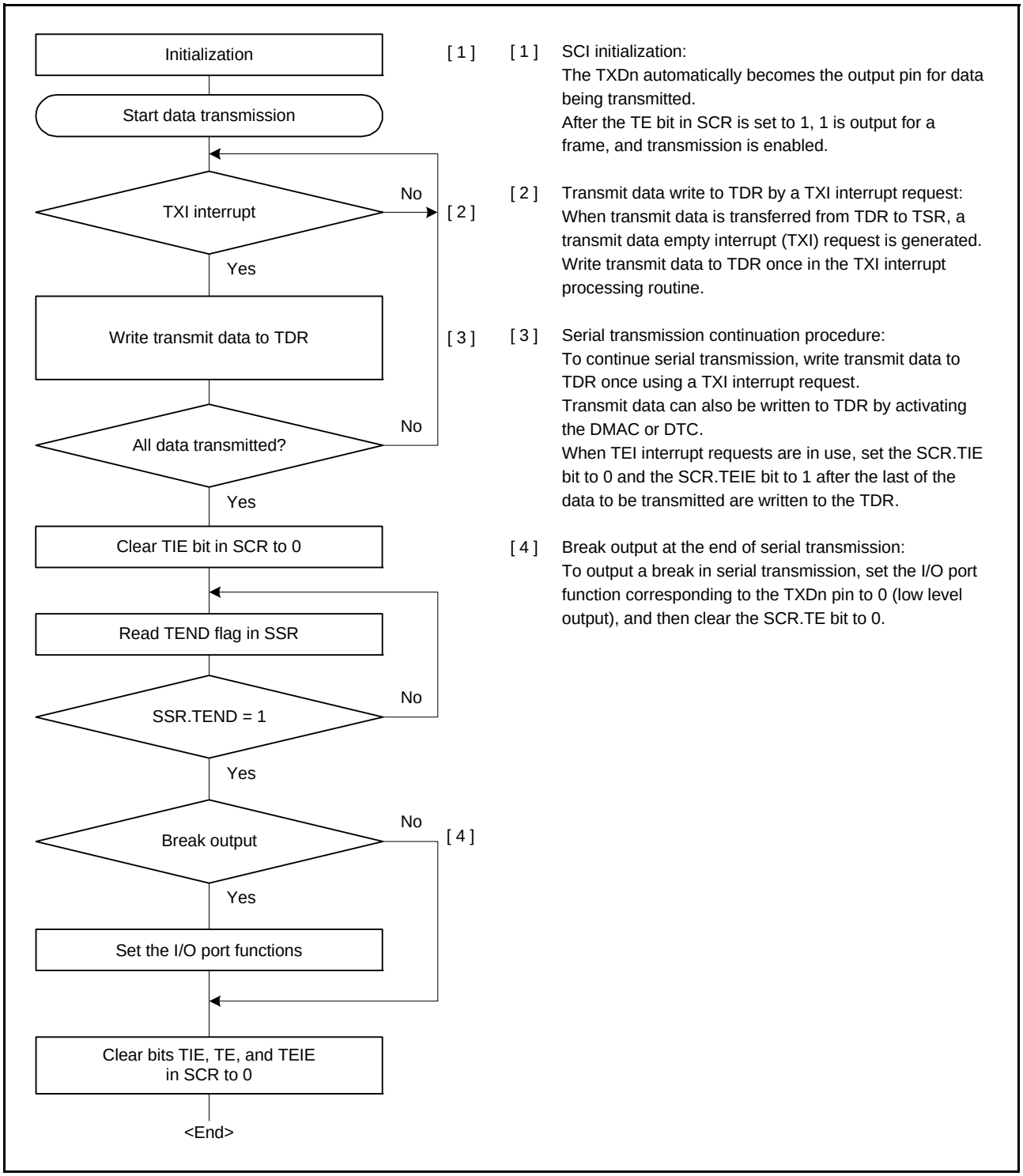
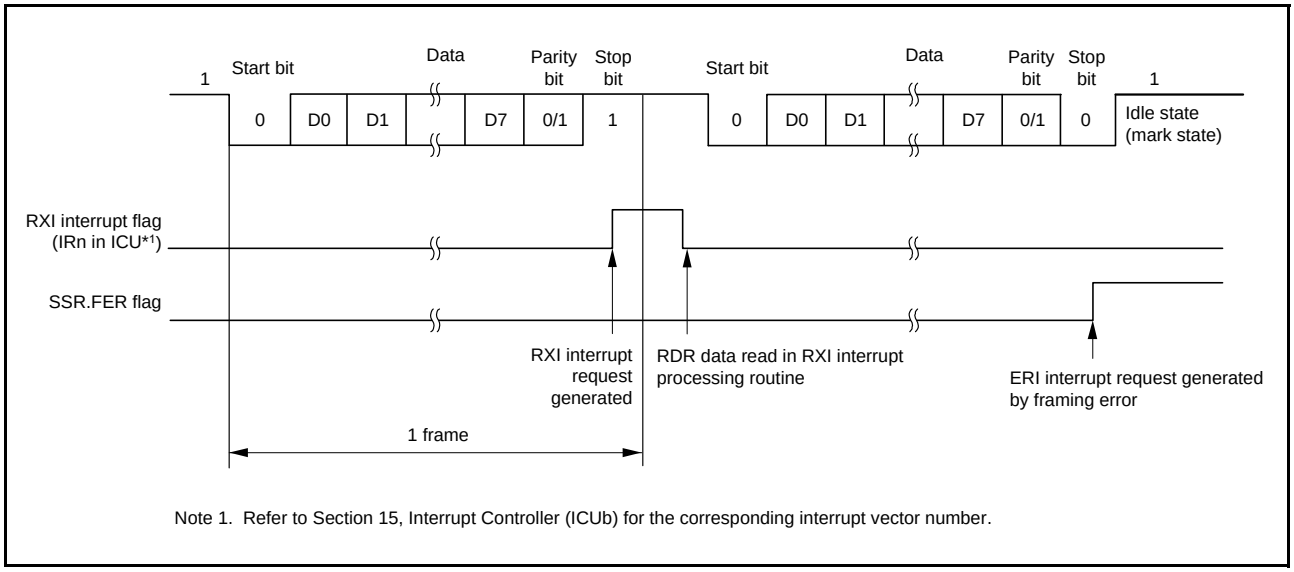


Figure 34.10 Example of Serial Transmission Flowchart in Asynchronous Mode

34.3.7 Serial Data Reception (Asynchronous Mode)

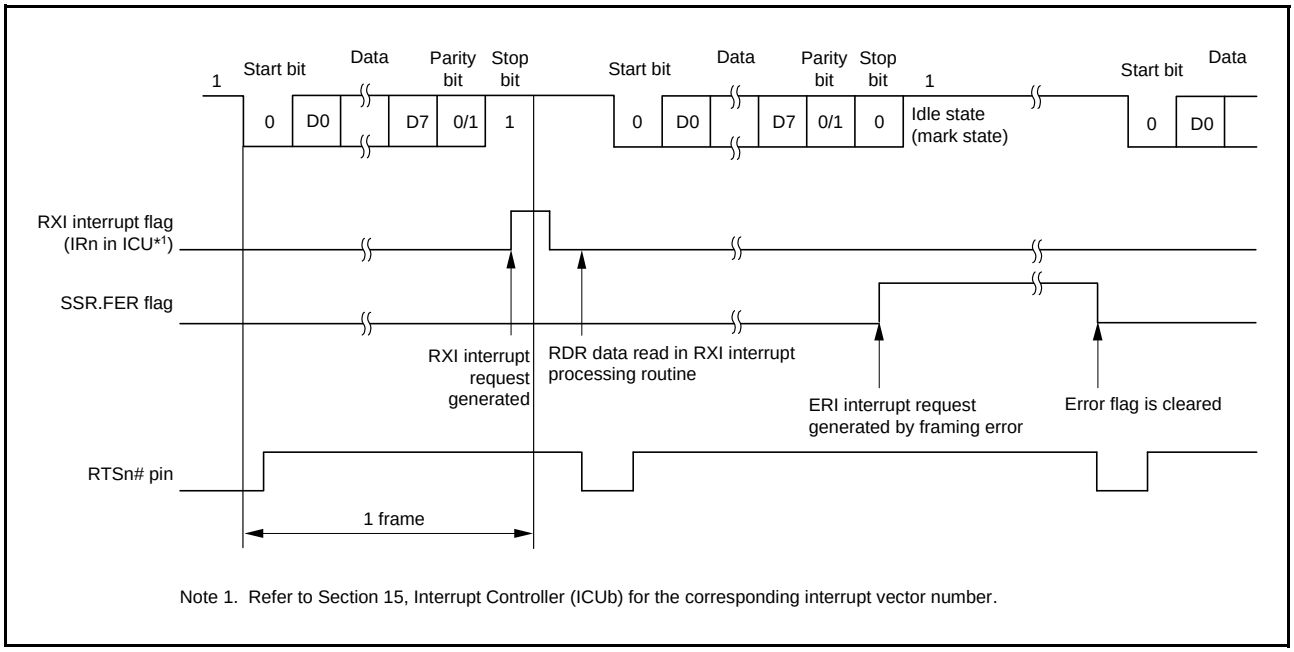
Figure 34.11 and Figure 34.12 shows an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

1. When the value of the RE bit in SCR 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 RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the ORER flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR 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 FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level.



Note 1. Refer to Section 15, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 34.11 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (when RTS Function is not Used) (Example with 8-Bit Data, Parity, One Stop Bit)



Note 1. Refer to Section 15, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 34.12 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (when RTS Function Is Used) (Example with 8-Bit Data, Parity, One Stop Bit)

Table 34.28 lists the states of the SSR status flags 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, clear the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing.
Figure 34.13 and Figure 34.14 show samples of flowcharts for serial data reception.

Table 34.28 SSR Status Flags and Receive Data Handling

SSR Status Flag				
ORER	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

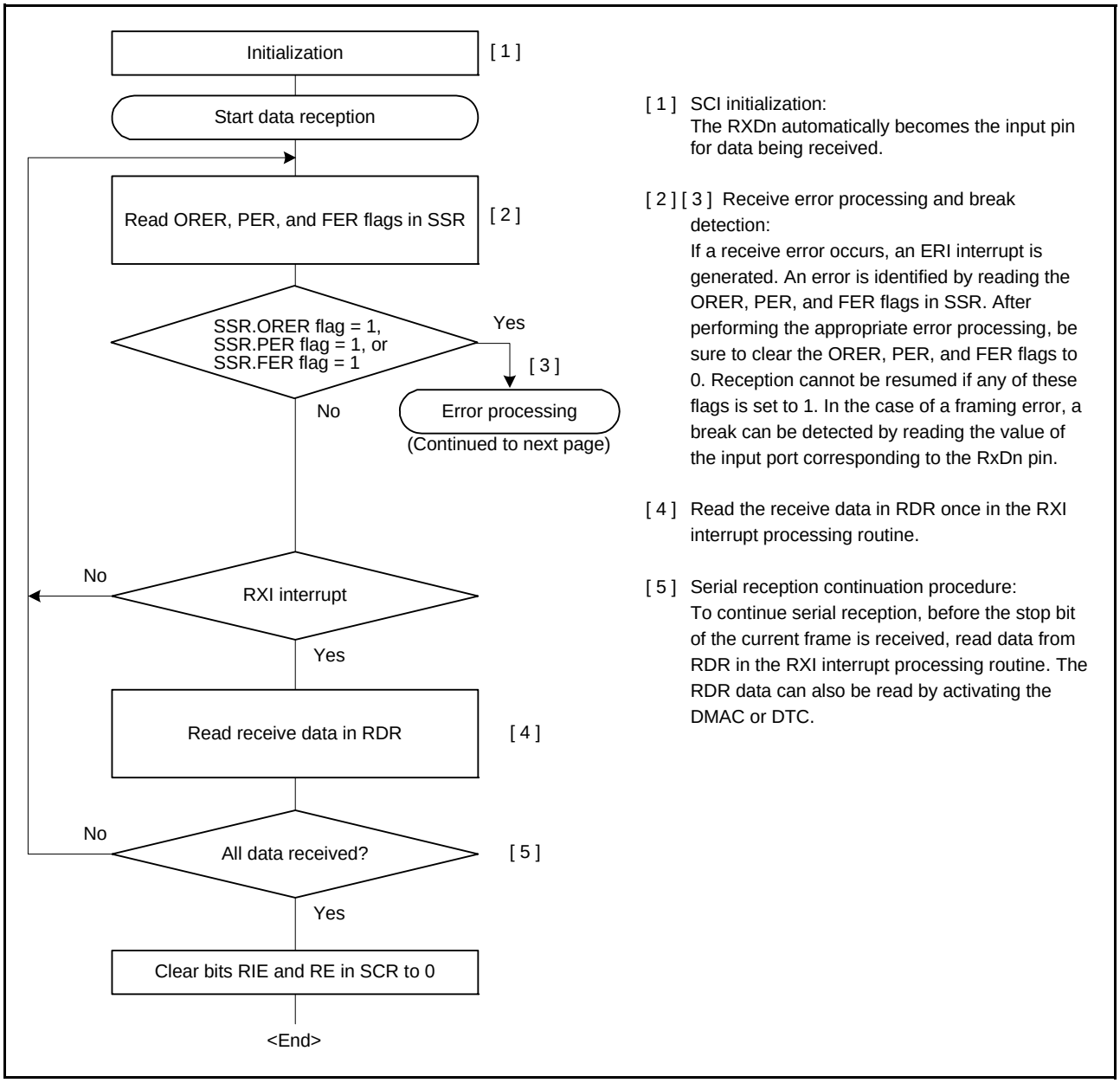


Figure 34.13 Example of Serial Reception Flowchart (1) (Asynchronous Mode)

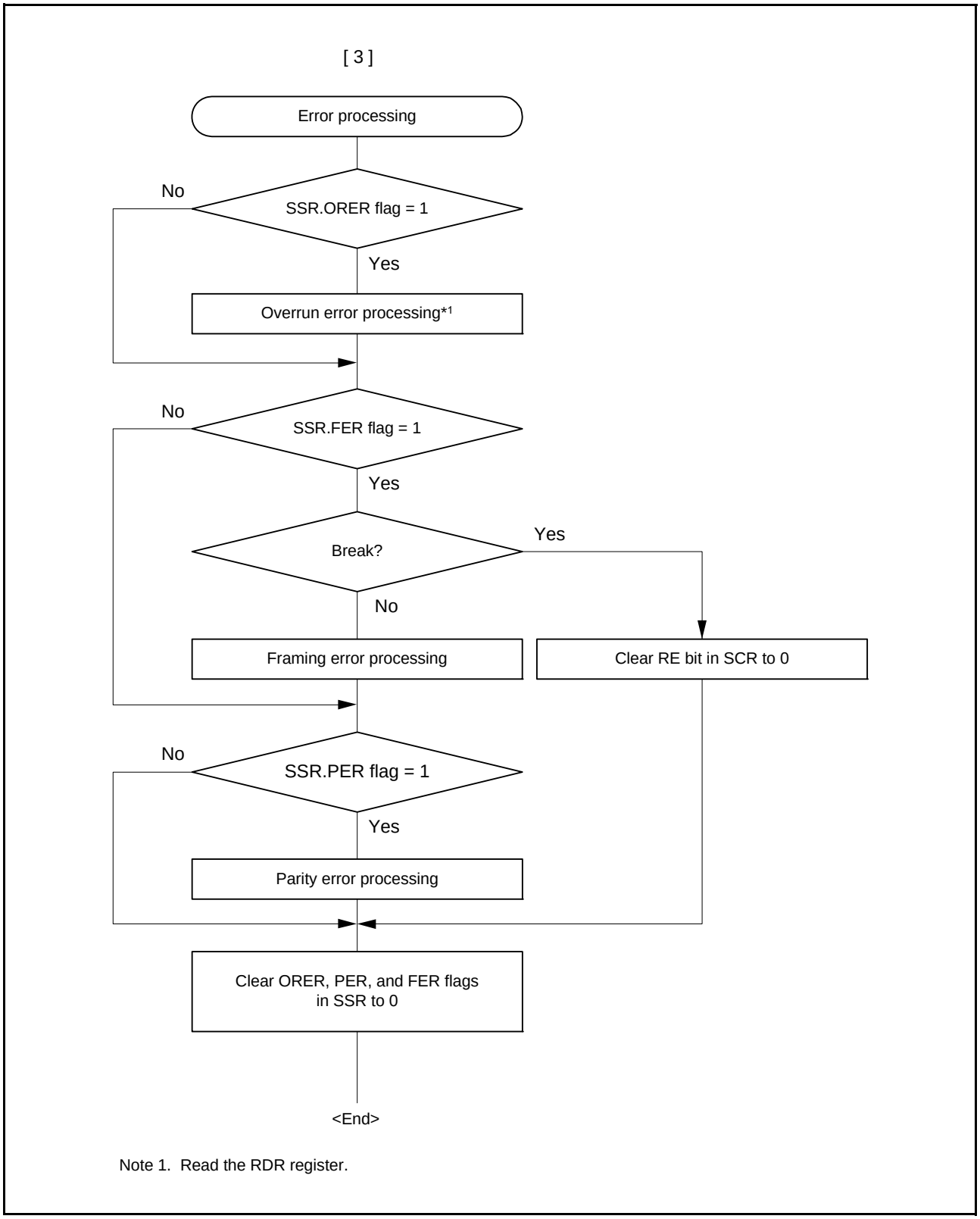


Figure 34.14 Example of Serial Reception Flowchart (2) (Asynchronous Mode)

34.4 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 34.15 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 transmission 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 MPIE bit in SCR. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR, detection of a reception error, and setting the respective status flags ORER and FER in SSR 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 MPBT bit in SSR is set to 1 and the MPIE bit in SCR is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the RIE bit in SCR is set.
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.

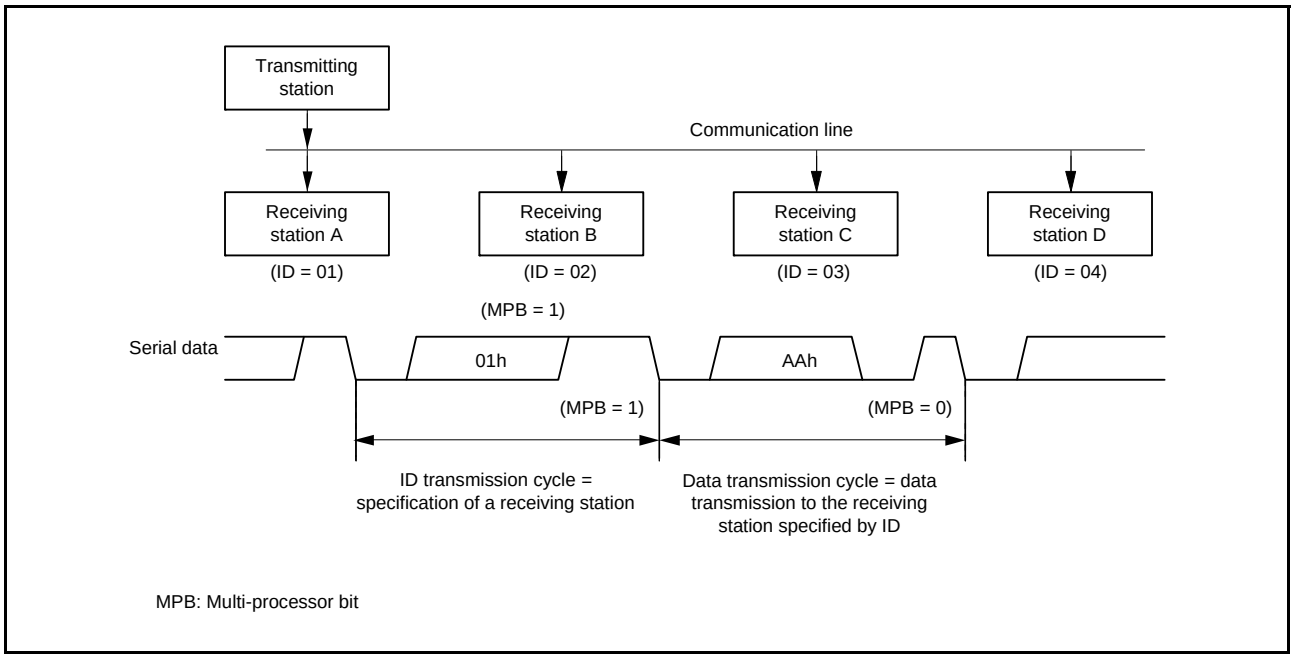


Figure 34.15 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

34.4.1 Multi-Processor Serial Data Transmission

Figure 34.16 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in SSR 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.

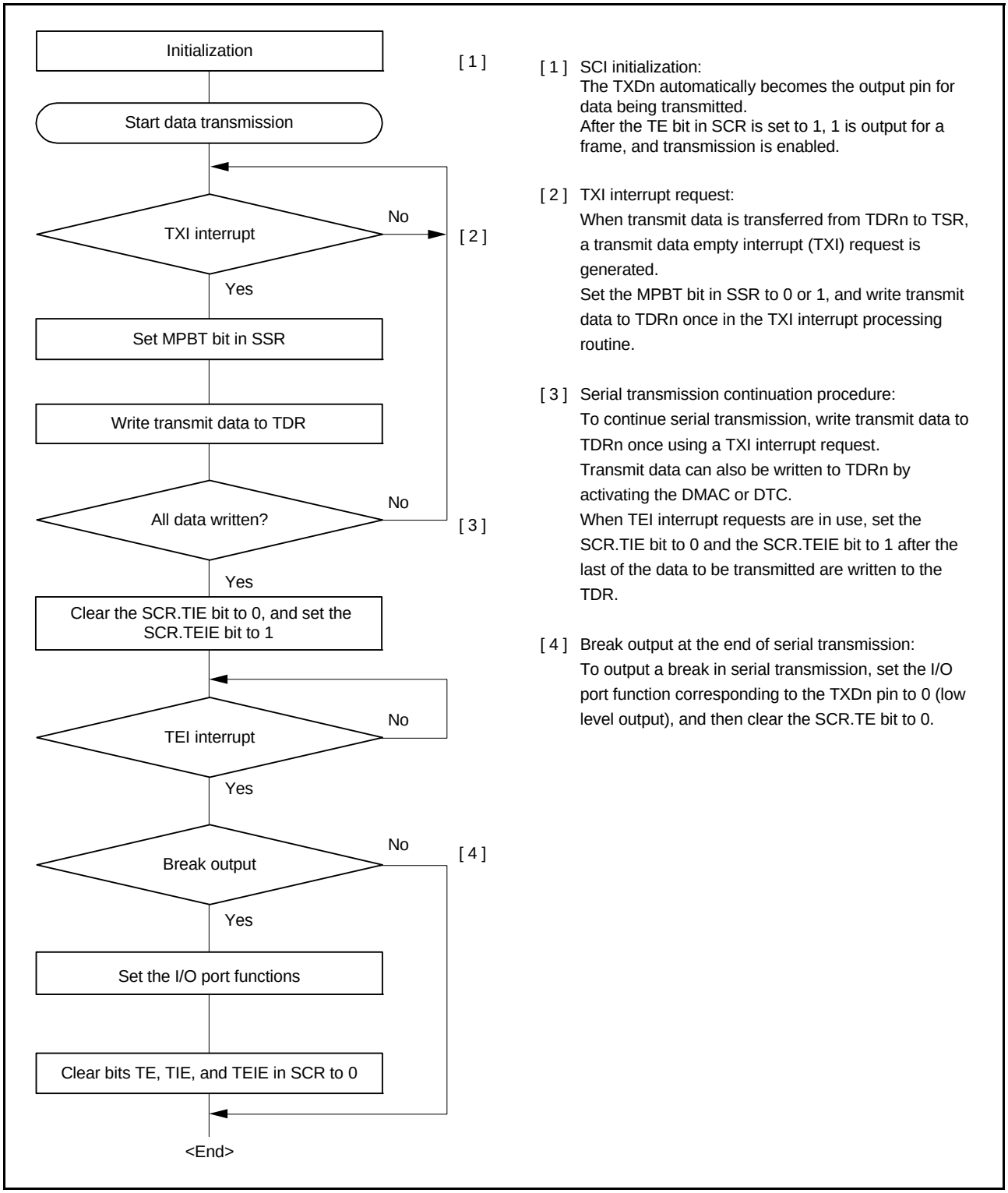


Figure 34.16 Example of Multi-Processor Serial Transmission Flowchart

34.4.2 Multi-Processor Serial Data Reception

Figure 34.18 and Figure 34.19 are sample flowcharts of multi-processor data reception. When the MPIE bit in SCR 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. During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 34.17 is the example of operation for reception.

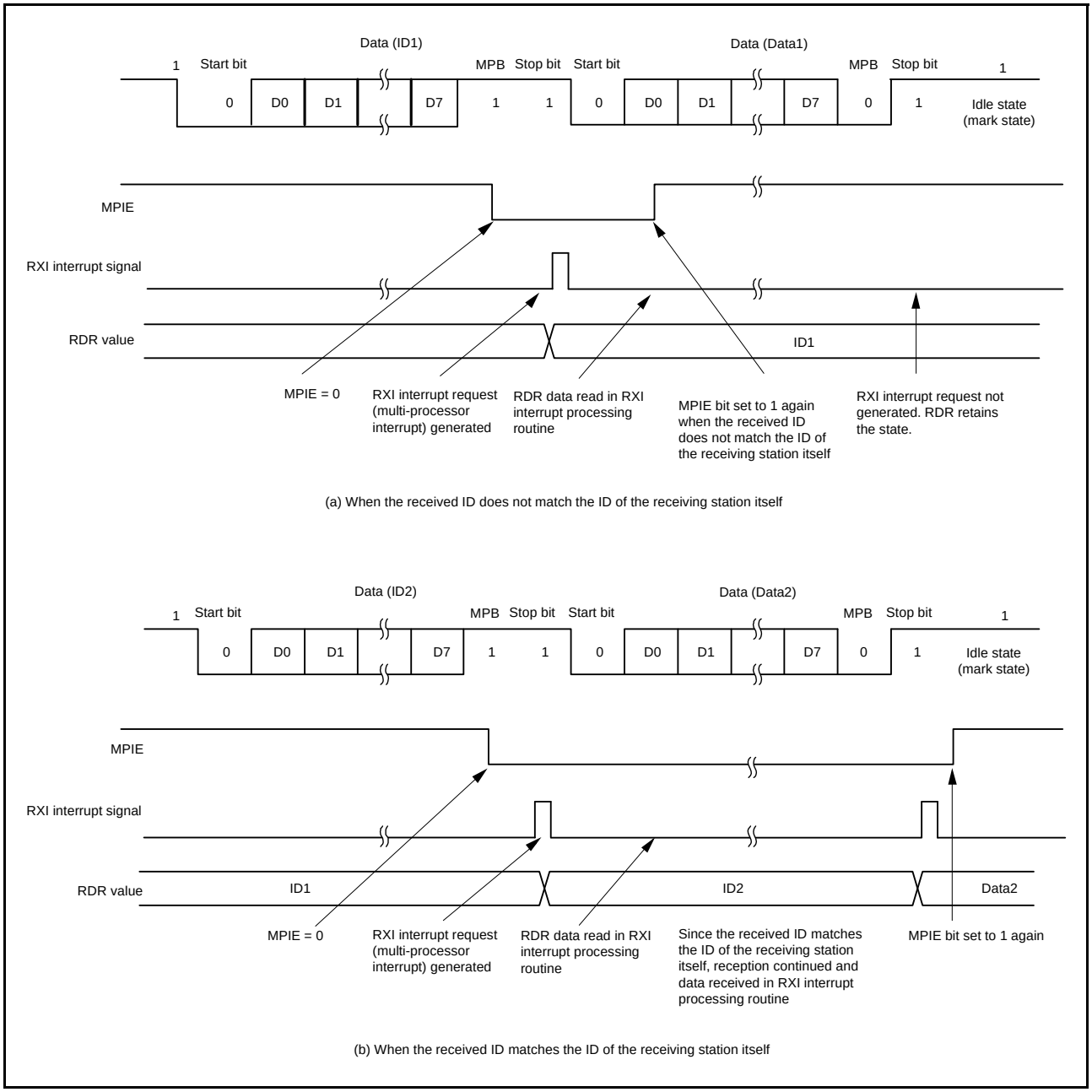


Figure 34.17 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/One Stop Bit)

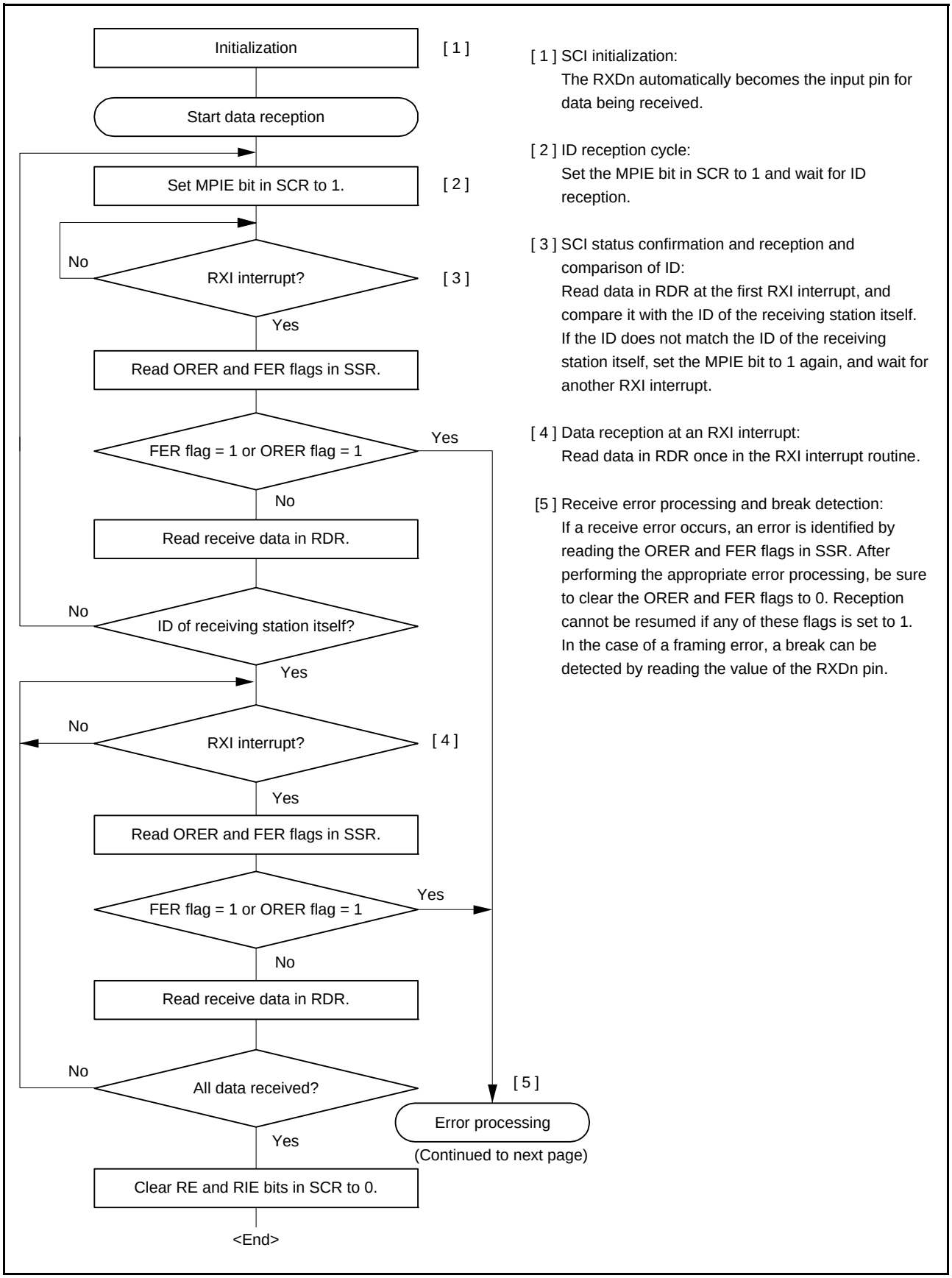


Figure 34.18 Example of Multi-Processor Serial Reception Flowchart (1)

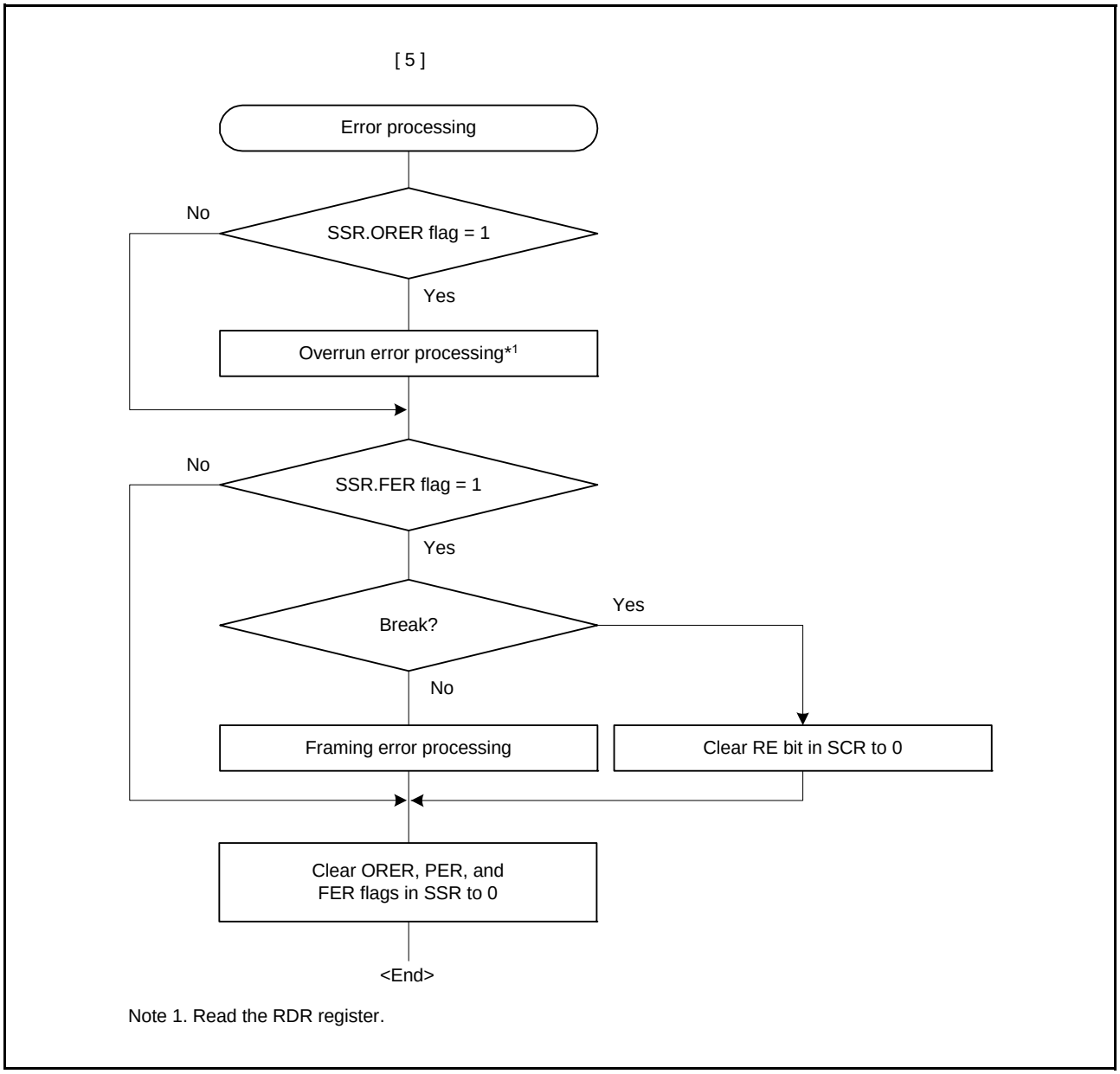
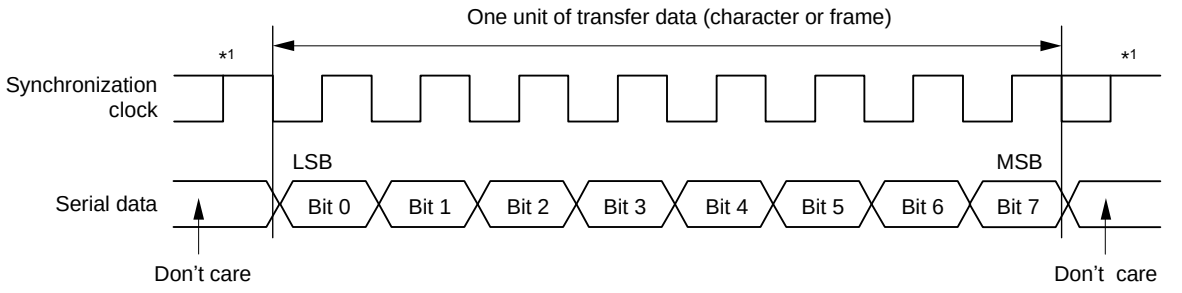


Figure 34.19 Example of Multi-Processor Serial Reception Flowchart (2)

34.5 Operation in Clock Synchronous Mode

Figure 34.20 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 transmission 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.



Note 1. Holds a high level except during continuous transfer.

Figure 34.20 Data Format in Clock Synchronous Serial Communications (LSB-First)

34.5.1 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 CKE[1:0] bits in SCR.
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, output of the synchronizing clock signal continues until the CTS function is enabled and the high level is input on the CTSn# pin, an overflow error occurs, or the RE bit in SCR is set to 0. When the CTS function is enabled, the synchronous clock signal output is stopped if the CTSn# pin input is high on completion of the frame reception.

34.5.2 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.
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]
Satisfaction of all conditions listed below

- The value of the RE or TE bit in the SCR is 1
- neither transmission nor reception is 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)
- ORER flag in SSR is 0

[Condition for high-level output]
Any of the conditions for the low level not being satisfied

34.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value “00h” to the SCR and then continue through the procedure for SCI given in the sample flowchart (Figure 34.21). Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.
Note that clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.
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.

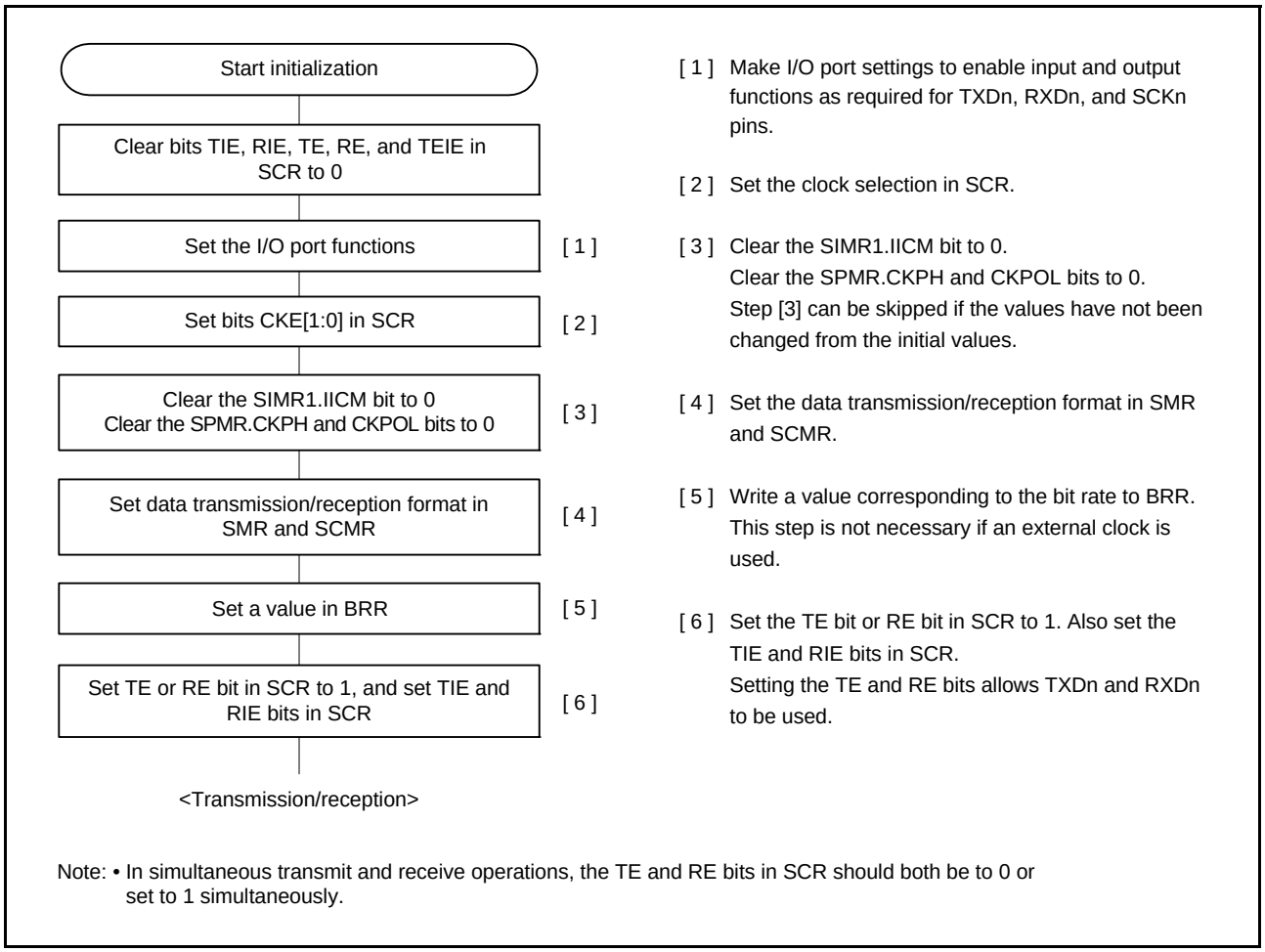


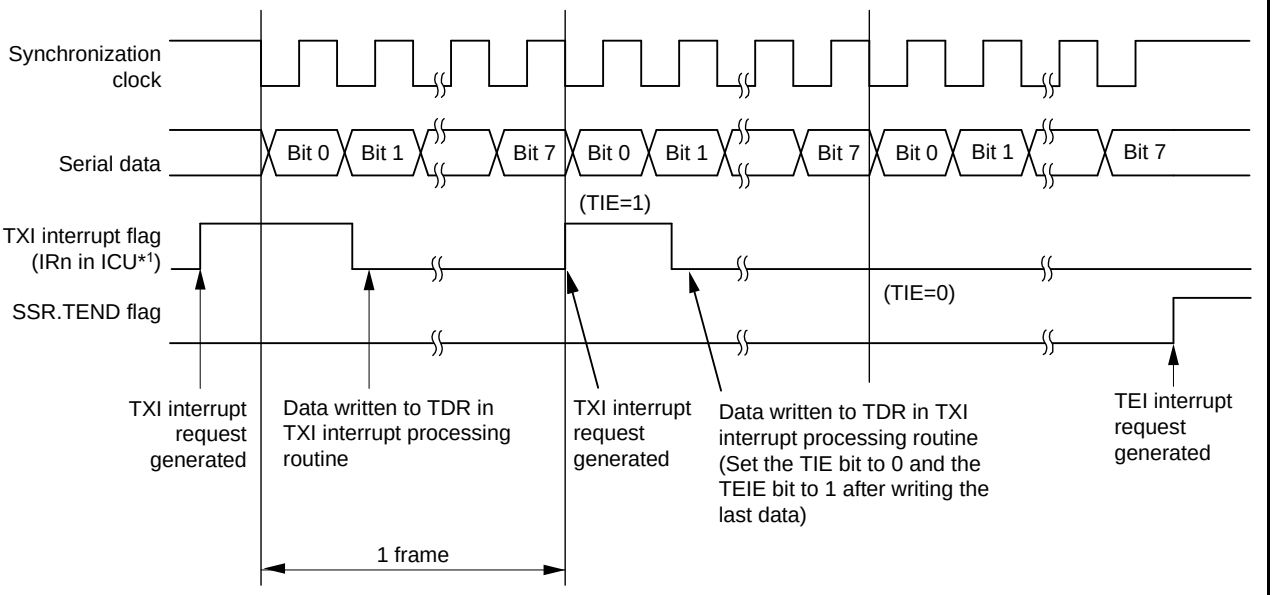
Figure 34.21 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

34.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 34.22 shows an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

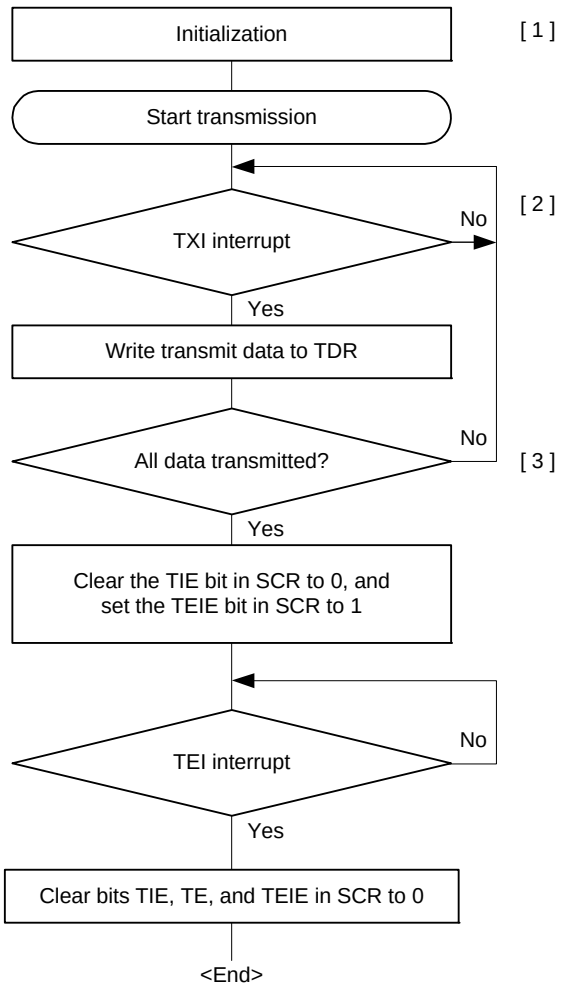
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, 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 TDR in this TXI interrupt processing routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (disabling TXI requests) and the SCR.TEIE bit to 1 (enabling TEI requests) after the last of the data to be transmitted are written to the TDR from the processing 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 SPMR is 1 (enabling the CTS function).
4. The SCI checks for updating of (writing to) the TDR at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If TDR is not updated, set the SSR flag in TEND to 1 and the TXDn pin retains the output state of the last bit. If the TEIE bit in SCR is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 34.23 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit in SCR to 0 does not clear the receive error flags.



Note 1. Refer to Section 15, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 34.22 Example of Operation for Serial Transmission in Clock Synchronous Mode (from the Middle of Transmission until Transmission Completion)



[1] SCI initialization:
The TXDn automatically becomes the output pin for data being transmitted.

[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 processing 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). Transmit data can also be written to TDR by activating the DMAC or DTC by the TXI interrupt request.

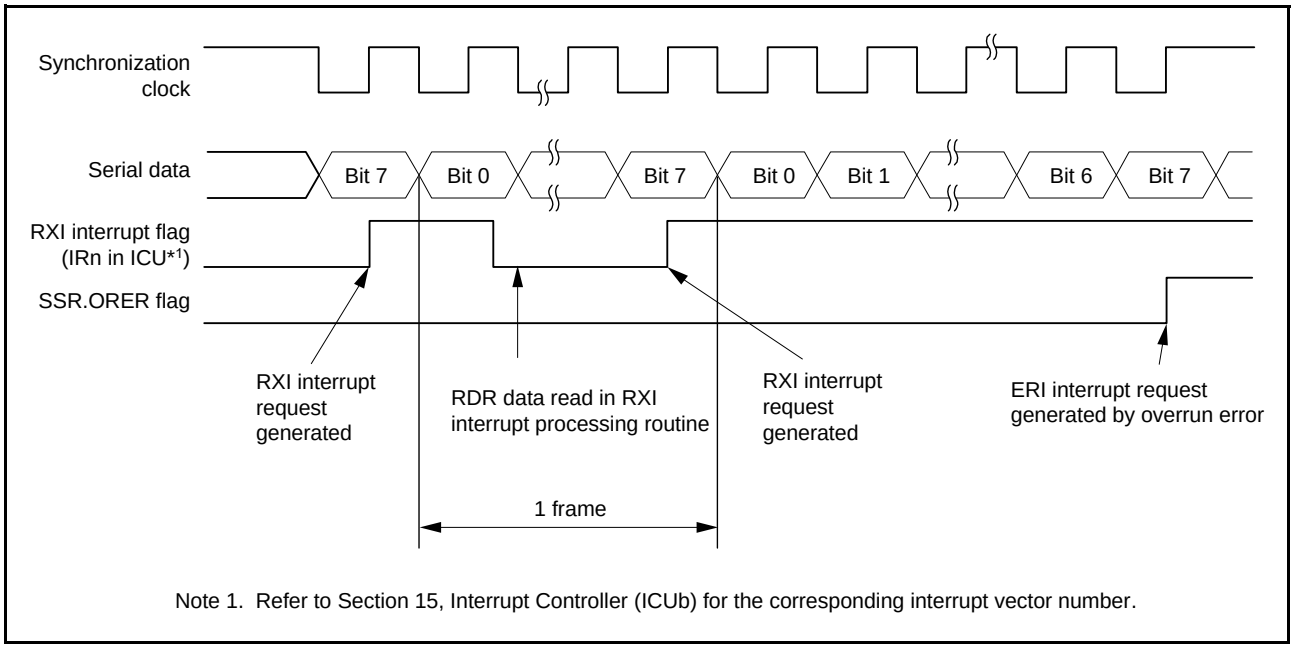
Note: • When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Clearing the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 34.23 Example of Serial Transmission Flowchart (Clock Synchronous Mode)

34.5.5 Serial Data Reception (Clock Synchronous Mode)

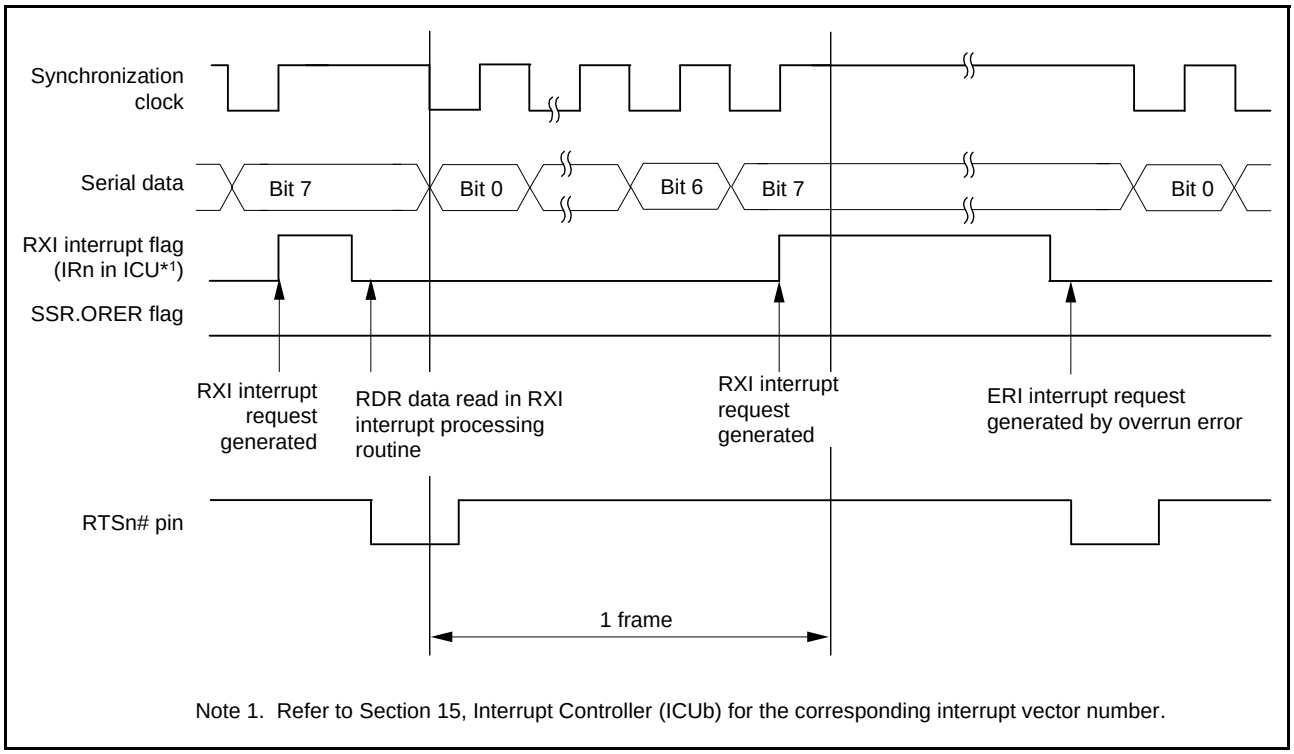
Figure 34.24 and Figure 34.26 shows examples of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the RE bit in SCR becoming 1 places the signal output on the RTS 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 RSR.
3. If an overrun error occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level (when the RTS function is in use).



Note 1. Refer to Section 15, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

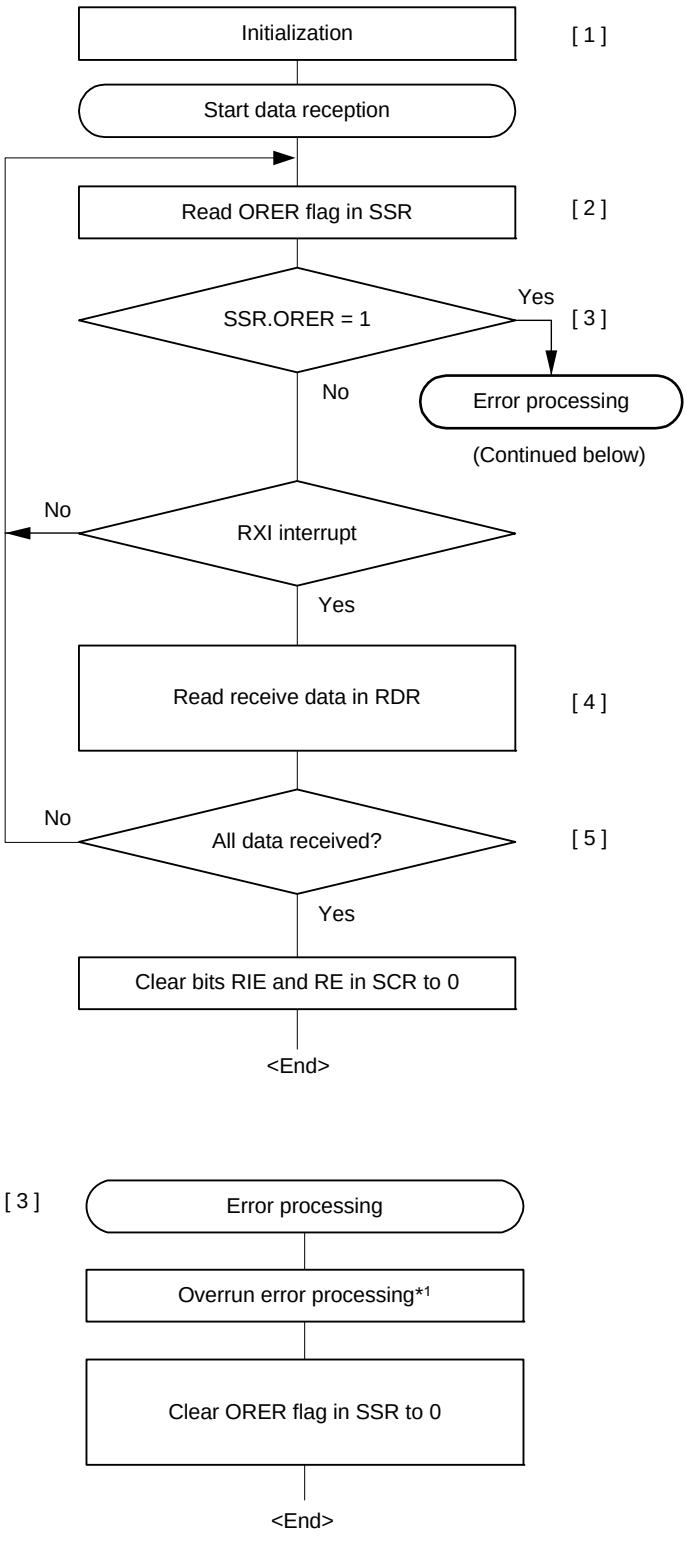
Figure 34.24 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (when RTS Function is not Used)



Note 1. Refer to Section 15, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 34.25 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 bits in SSR to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing.
Figure 34.26 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 SSR, perform the relevant error processing, and then clear the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Read the receive data in RDR once in the receive data full interrupt (RXI) request processing 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 RDR. The RDR data can also be read by activating the DMAC or DTC by an RXI interrupt request.

Note 1. Read the RDR register.

Figure 34.26 Example of Serial Reception Flowchart (Clock Synchronous Mode)

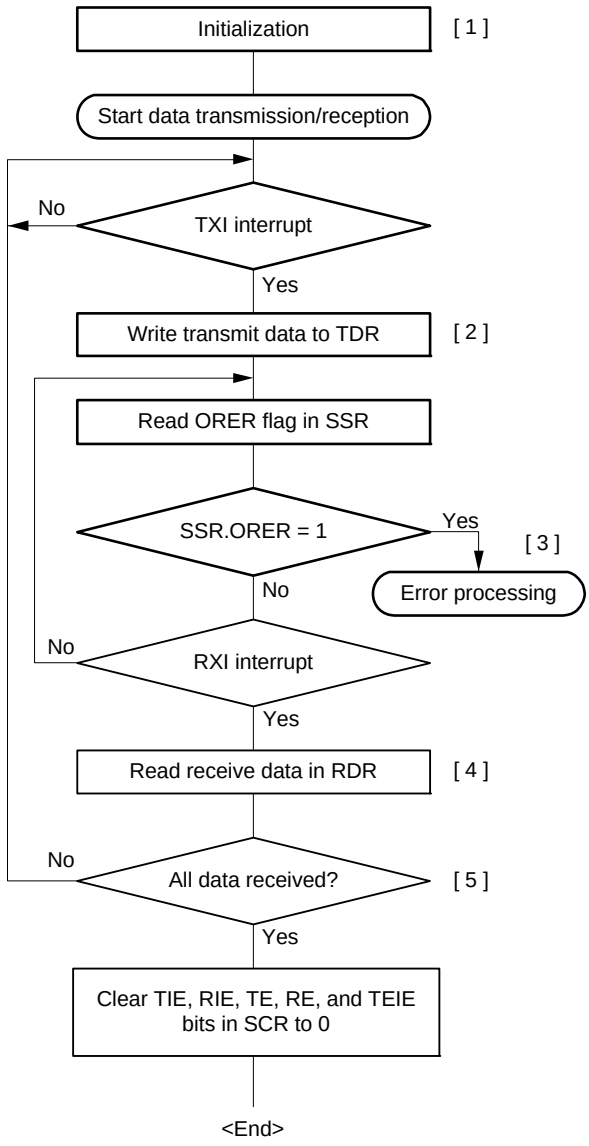
34.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 34.27 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 SSR is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in SCR 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 clear the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in SSR) are 0, and then set the TIE, RIE, TE, and RE bits in SCR 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 request processing routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, and then clear 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 RDR once in the RXI interrupt request processing 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 RDR by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to TDR by the TXI interrupt.
Transmit data can also be written to TDR 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 clear 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 34.27 Example of Simultaneous Serial Transmission and Reception Flowchart (Clock Synchronous Mode)

34.6 Operation in Smart Card Interface Mode

The SCI supports the smart card (IC card) interface conforming to the ISO/IEC 7816-3 (Identification Card) standard, as an extended function of the SCI.
Smart card interface mode can be selected using the appropriate register.

34.6.1 Sample Connection

Figure 34.28 shows a sample connection between a smart card (IC card) and this LSI.
As in the figure, since this LSI 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 SCR 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 RX63N/RX631 can be used to output a reset signal.

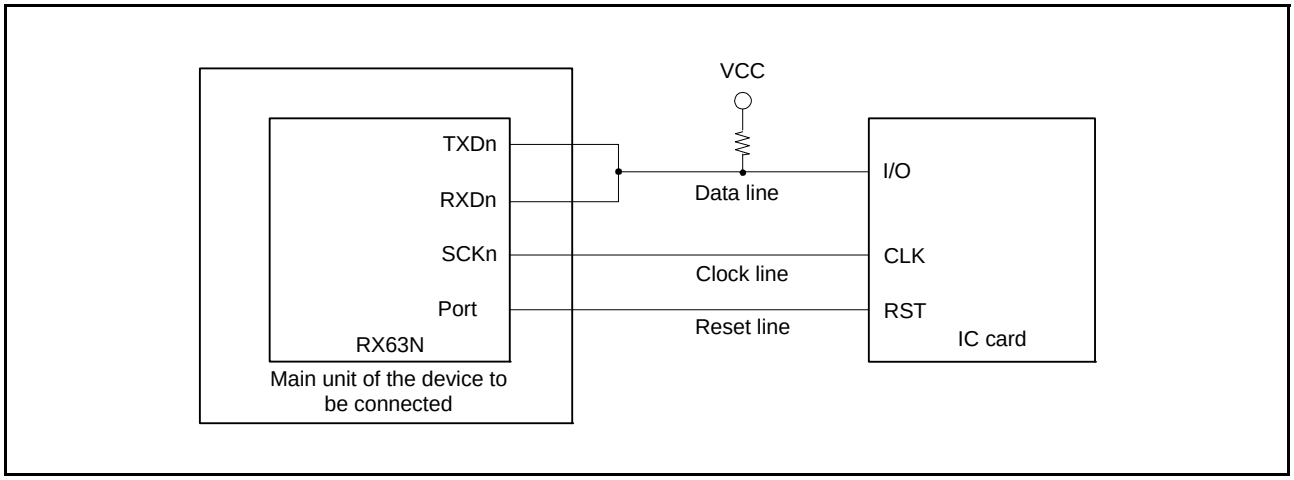


Figure 34.28 Sample Connection with a Smart Card (IC Card)

34.6.2 Data Format (Except in Block Transfer Mode)

Figure 34.29 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 one 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 re-transmitted after at least 2 etu.

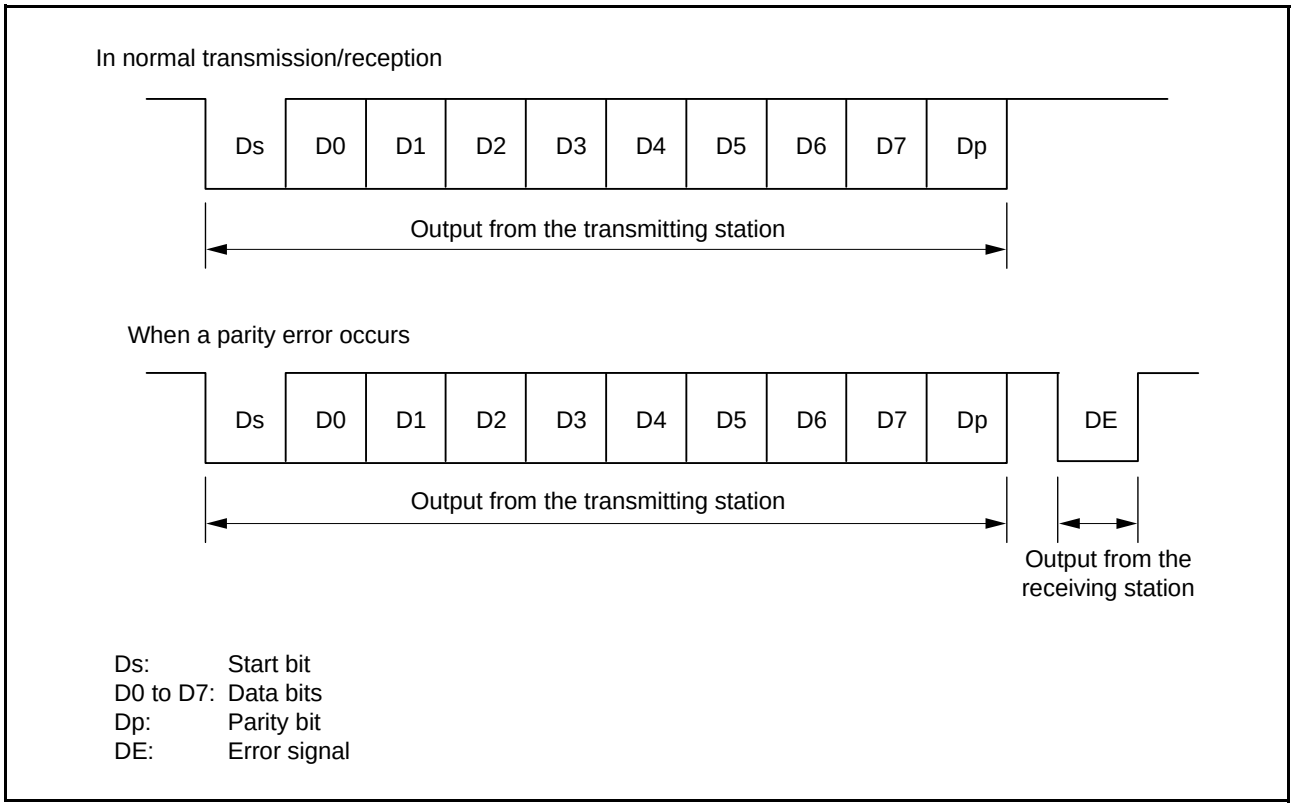


Figure 34.29 Data Formats in Smart Card Interface Mode

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 34.30. 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 SCMR. Write 0 to the PM bit in SMR in order to use even parity, which is prescribed by the smart card standard.

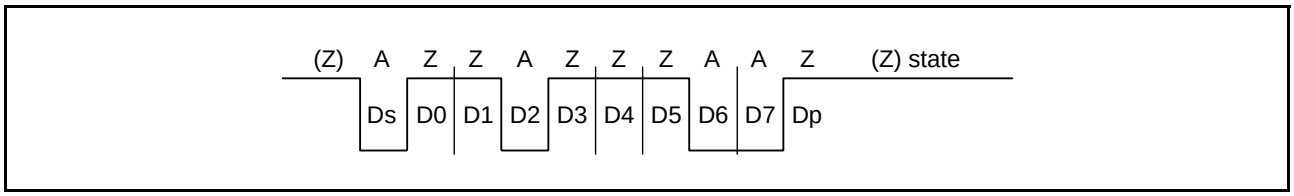


Figure 34.30 Direct Convention (SDIR in SCMR = 0, SINV in SCMR =0, PM in SMR = 0)

(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 34.31. 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 SCMR. 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 RX63N/RX631 only inverts data bits D7 to D0, write 1 to the PM bit in SMR to invert the parity bit for both transmission and reception.

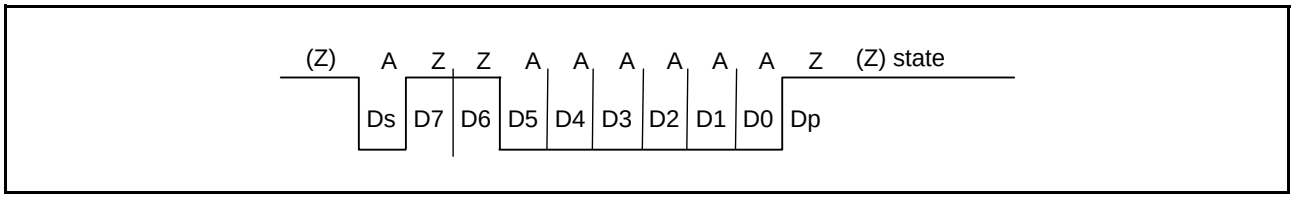


Figure 34.31 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR =1, PM in SMR = 1)

34.6.3 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 SSR 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 re-transmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in SSR indicates the error signal status as in normal smart card interface mode, but the flag is always read as 0 because no error signal is transferred.

34.6.4 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 SCMR and the BCP[1:0] bits in SMR (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 34.32. 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.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$$

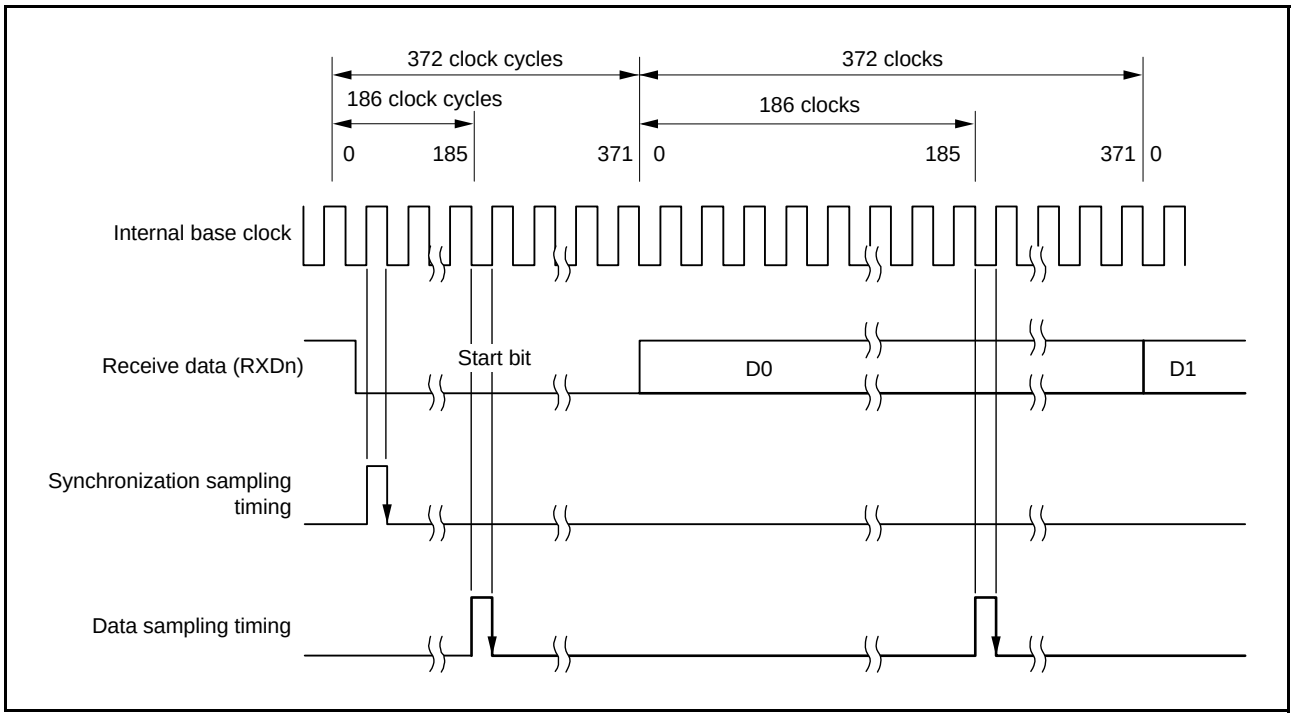


Figure 34.32 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

34.6.5 Initialization of the SCI (Smart Card Interface Mode)

Before transmitting and receiving data, initialize the SCI using the following procedure. Initialization is also necessary before switching from transmission to reception and vice versa.

1. Write the initial value “00h” to the SCR.
2. Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
3. Set the error flags ORER, ERS, and PER in SSR to 0.
4. Clear the SIMR1.IICM bit and the SPMR.CKPH and CKPOL bits to 0.
   This step can be skipped if the values have not been changed from the initial values.
5. Set bits GM, BLK, OE, BCP[1:0], and CKS[1:0] in SMR and the BCP2 bit in SCMR appropriately. Also set the PE bit in SMR to 1.
6. Set bits SDIR, SINV, and SMIF in SCMR appropriately. Then, the TXDn and RXDn pins are placed in the high impedance state.
7. Set the value corresponding to the bit rate in BRR.
8. Set the CKE[1:0] bits in SCR appropriately, and set bits TIE, RIE, TE, RE, and TEIE in SCR to 0 at the same time. When the CKE[1:0] bit is set to 1, the SCKn pin is allowed to output clock pulses.
9. Set the TIE, RIE, TE, and RE bits in SCR to 1. Setting the TE and RE bits to 1 simultaneously is prohibited except for self diagnosis.

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 SSR.
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 SSR.

34.6.6 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 re-transmitted, is different from that in normal serial communications interface mode. Figure 34.33 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 SSR is set to 1. If the RIE bit in SCR 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 SSR is not set. Data is retransferred from TDR to TSR 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 SCR is 1 at this time, a TXI interrupt request is generated. Writing transmit data to TDR starts transmission of the next data.

Figure 34.35 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 SSR is set to 1 in transmission, if the TIE bit in SCR 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 cleared to 0 when the DTC or DMAC transfers the data.
If an error occurs, the SCI automatically re-transmits 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, see section 17, DMA Controller (DMACA) and section 19, Data Transfer Controller (DTCa).

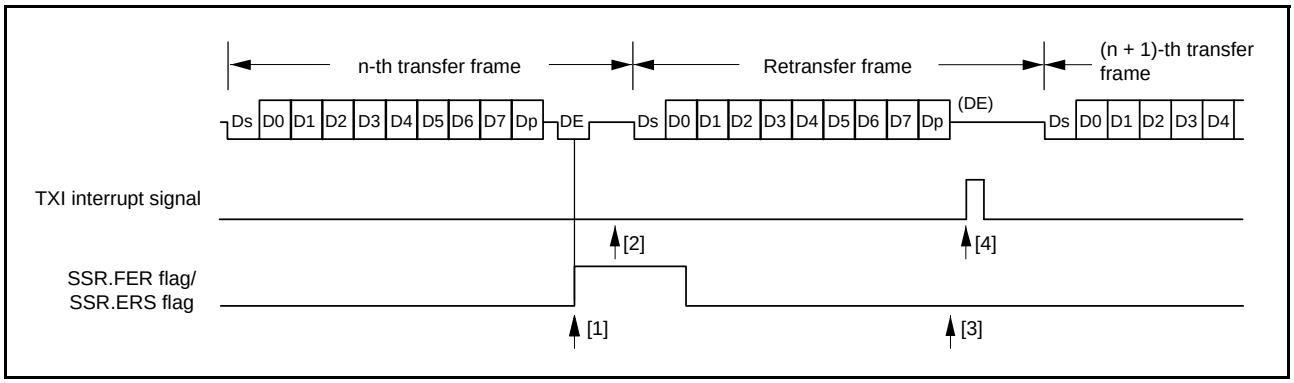


Figure 34.33 Data Retransfer Operation in SCI Transmission Mode

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 34.34 shows the TEND flag generation timing.

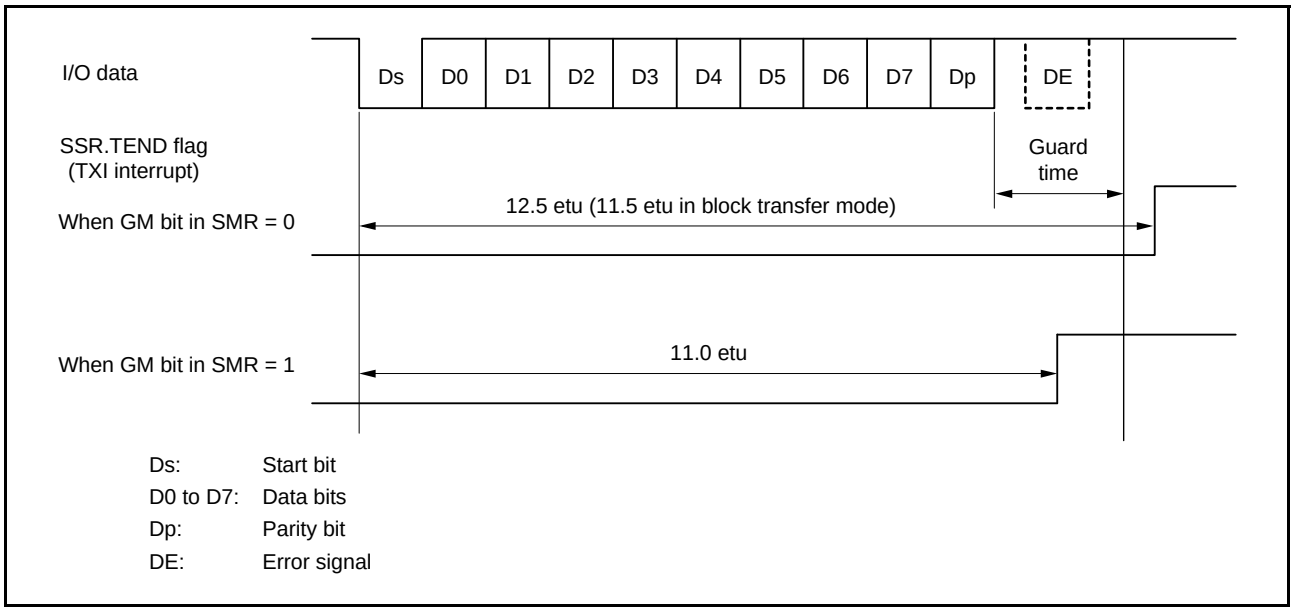


Figure 34.34 SSR.TEND Flag Generation Timing during Transmission

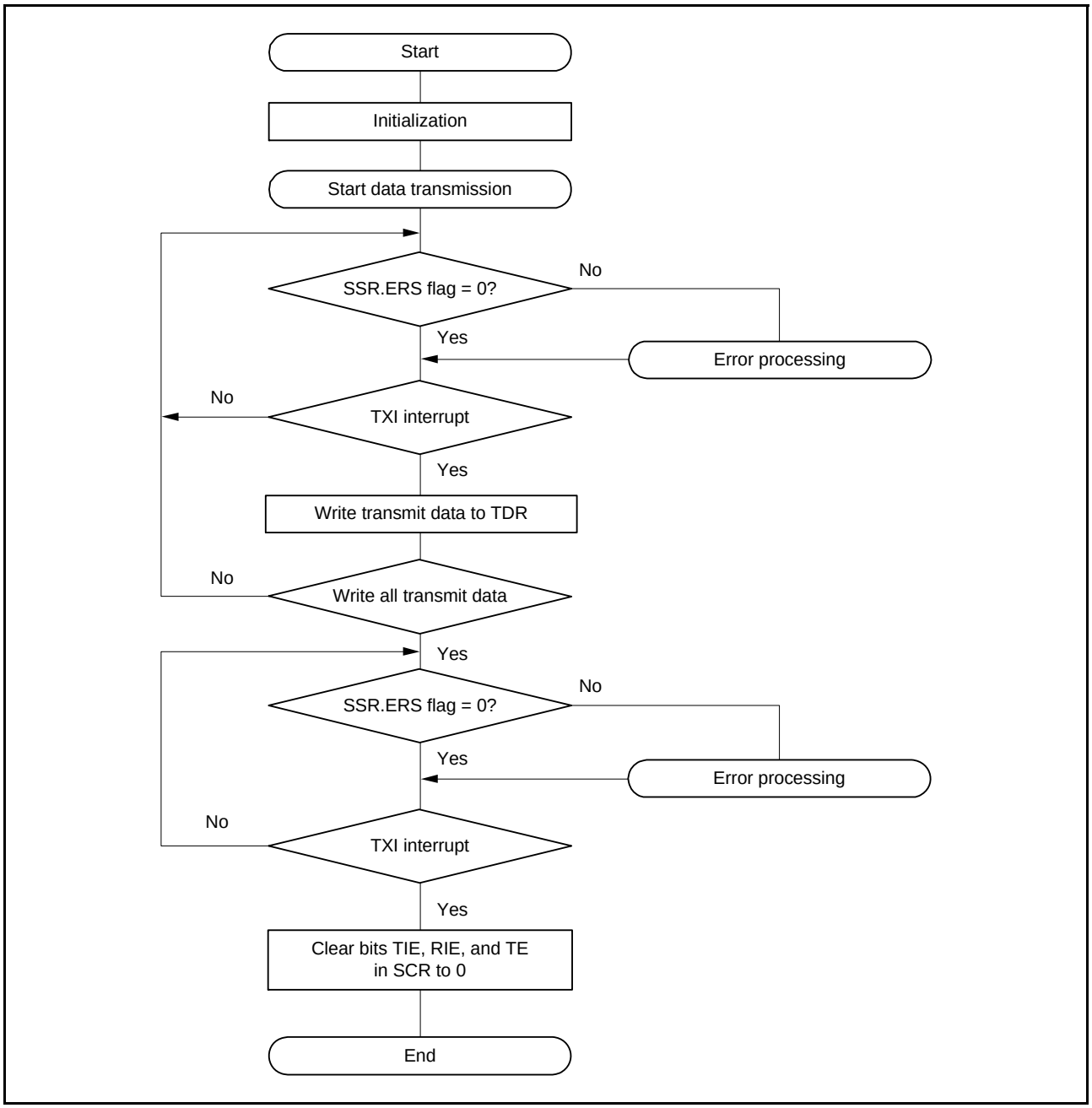


Figure 34.35 Sample Smart Card Interface Transmission Flowchart

34.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in serial communications interface mode. Figure 34.36 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the PER flag in SSR is set to 1. When the RIE bit in SCR 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 SSR is not set to 1.
4. In this case, data is determined to have been received successfully. When the RIE bit in SCR is 1, an RXI interrupt request is generated.

Figure 34.37 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 SSR 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.
Note 1. For operations in block transfer mode, see section 34.3, Operation in Asynchronous Mode.

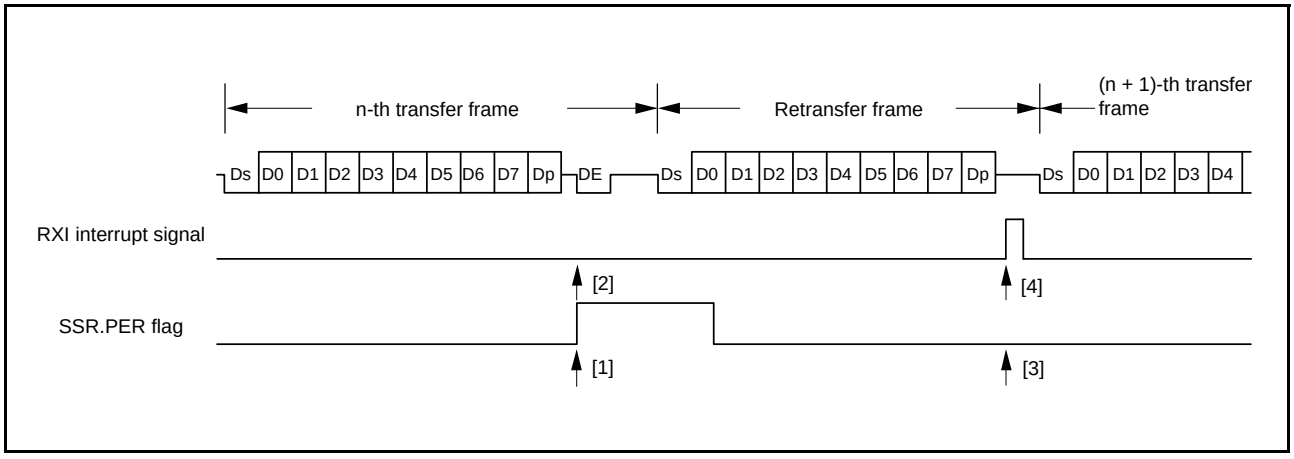


Figure 34.36 Data Retransfer Operation in SCI Reception Mode

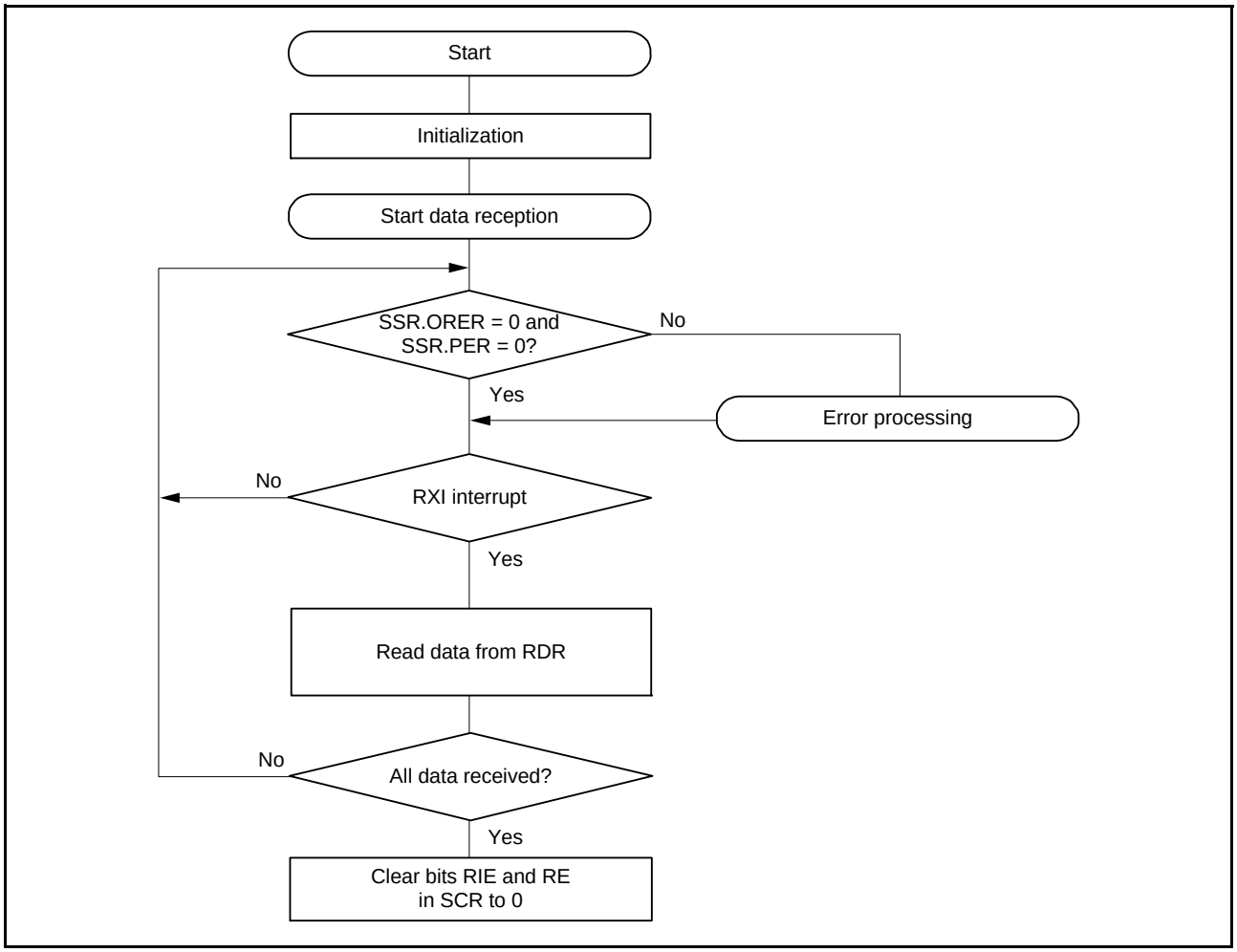


Figure 34.37 Sample Smart Card Interface Reception Flowchart

34.6.8 Clock Output Control

Clock output can be fixed using the CKE[1:0] bits in SCR when the GM bit in SMR is 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 34.38 shows an example of clock output fixing timing when the CKE0 bit is controlled with GM = 1 and CKE1 = 0.

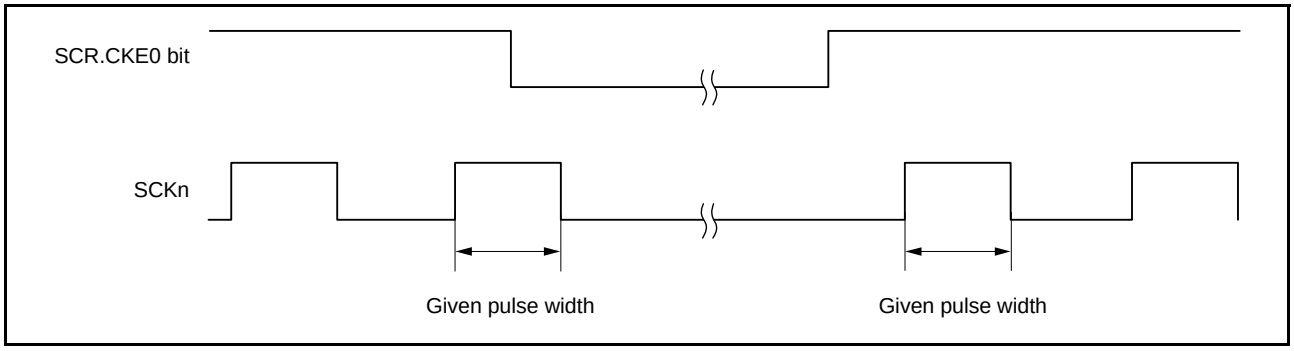


Figure 34.38 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedure to secure the appropriate clock duty cycle.

(1) At Power-On

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.

(2) At Mode Switching

(a) At transition from smart card interface mode to software standby mode

1. Set I/O port functions to make the SCKn pin fixed with a desired output value in software standby mode.
2. Write 0 to the TE and RE bits in SCR to stop transmission/reception.
   Simultaneously, set the SCR.CKE[1] bit to the value for the output fixed state in software standby mode.
3. Write 0 to the SCR.CKE[0] bit to stop the clock.
4. Wait for one cycle of the serial clock. In the mean time, the clock output is fixed to the specified level with the duty cycle retained.
5. Make a transition to software standby mode.

(b) Return from software standby mode to smart card interface mode

6. Cancel software standby mode.
7. Set the SCR.CKE[0] bit to 1 to start clock output. A clock signal with the appropriate duty cycle is then generated.

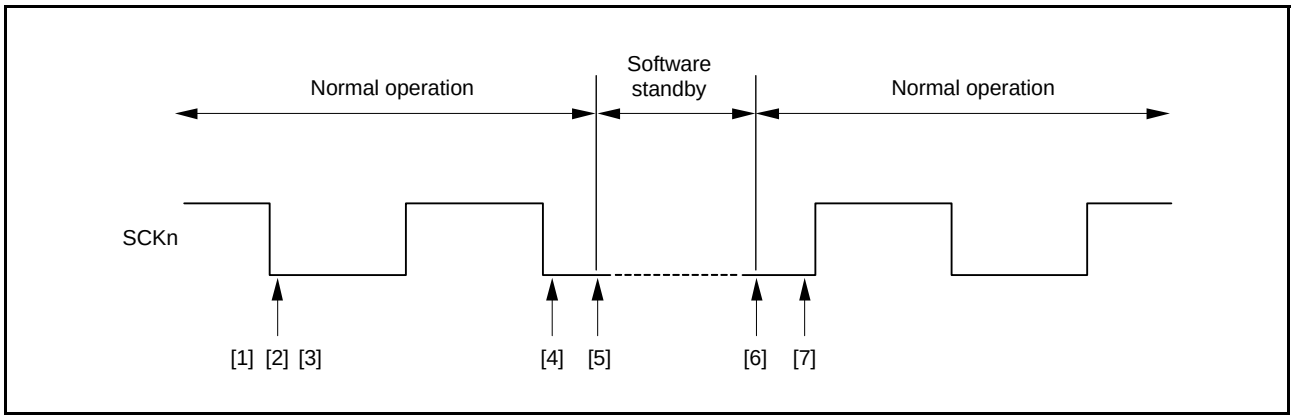


Figure 34.39 Clock Stop and Restart Procedure

34.7 Operation in Simple I²C Mode

Simple I²C bus master format is composed of eight 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 eight 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 34.40 and Figure 34.41.

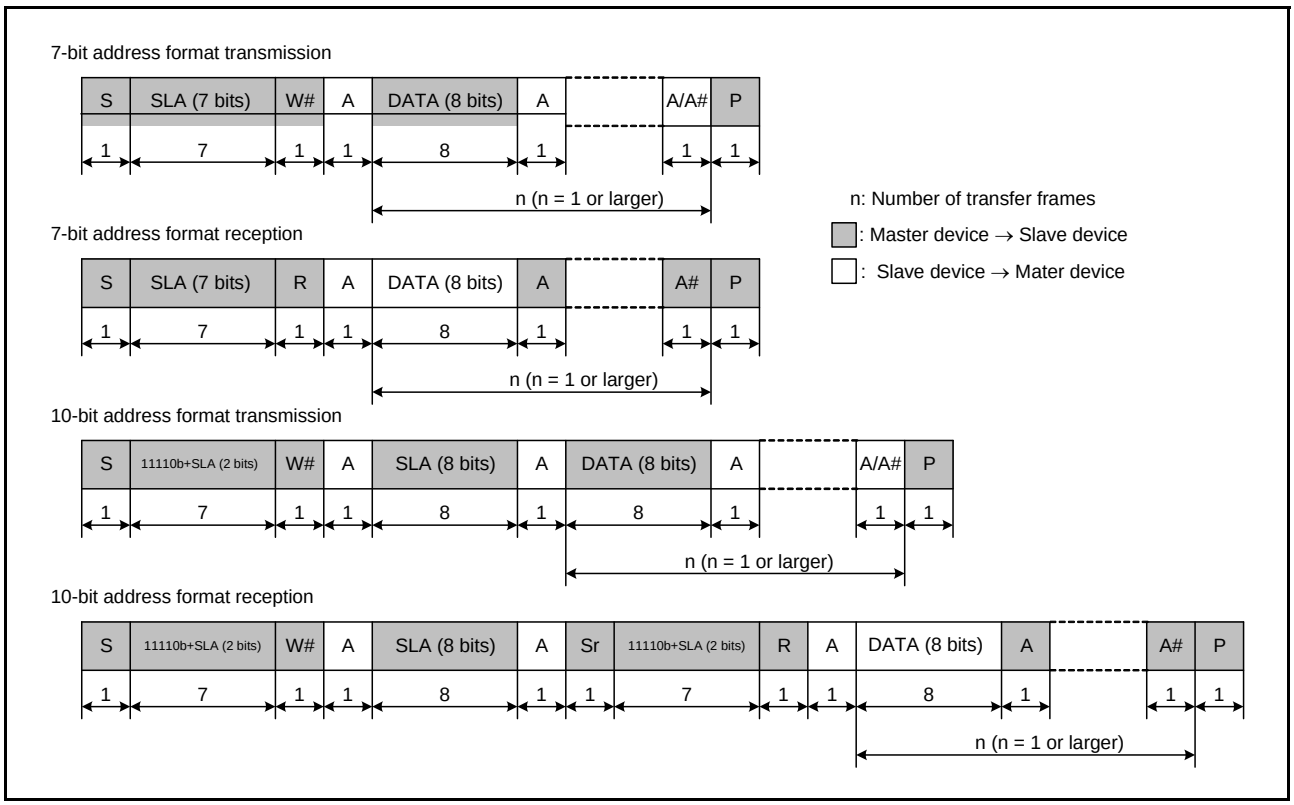


Figure 34.40 I²C Bus Master Format

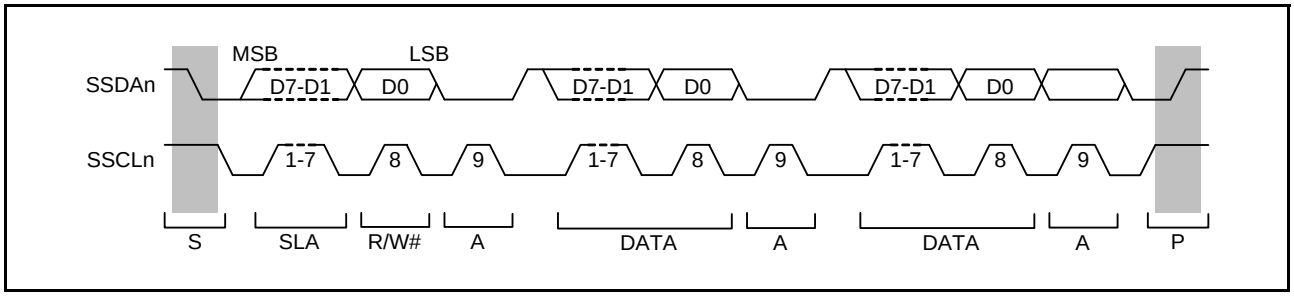


Figure 34.41 I²C Bus Timing (When SLA is 7 Bits)

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.

34.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the IICSTAREQ bit in SIMR3 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 SIMR3 is cleared (to 0), and a start-condition generated interrupt is output.

Writing 1 to the IICRSTAREQ bit in SIMR3 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 SIMR3 is cleared (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the IICRSTPREQ bit in SIMR3 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 SIMR3 is cleared (to 0), and a stop-condition generated interrupt is output.

Figure 34.42 shows the timing of operations in the generation of start, restart, and stop conditions.

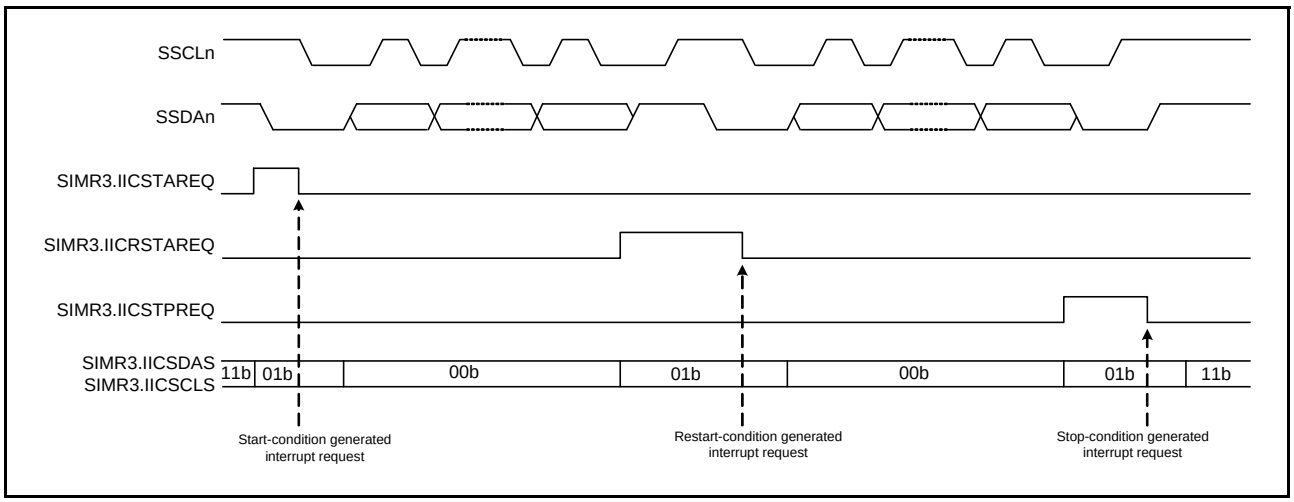


Figure 34.42 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

34.7.2 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 SIMR2 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 SIMR2 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 input, 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 SIMR2 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 SIMR2 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 34.43 shows an example of operations to synchronize the clocks.

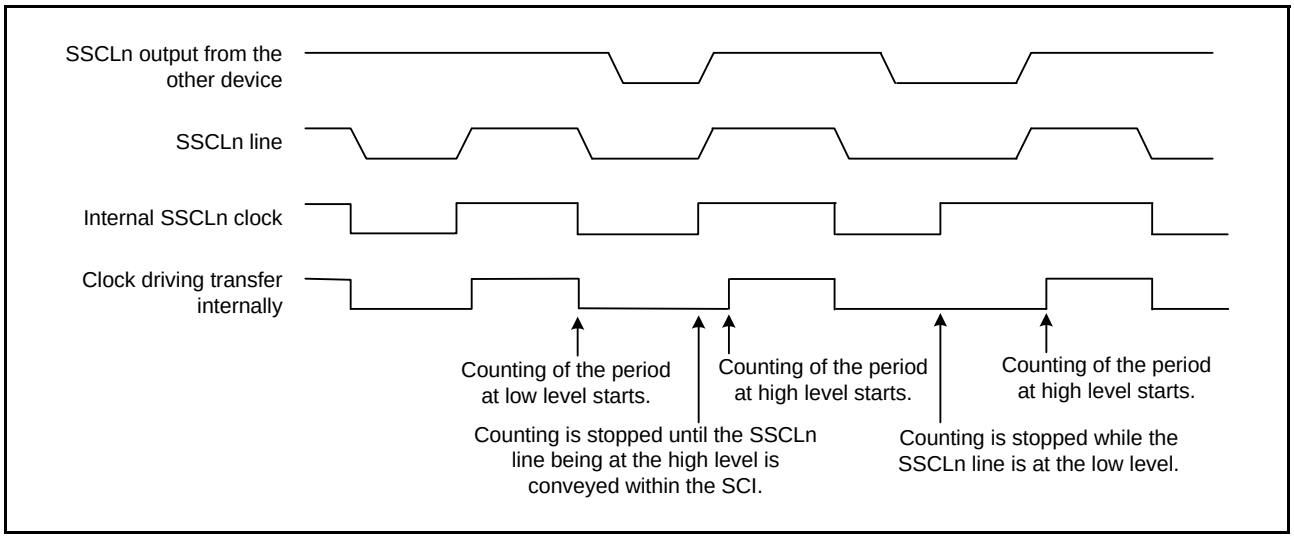


Figure 34.43 Example of Operations for Clock Synchronization

34.7.3 SSDA Output Delay

The IICDL[4:0] bits in SIMR1 can be used to set a delay for output on the SSDA 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 SMR). 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 SSDA 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 34.44 shows the timing of delays in output on the SSDA pin.

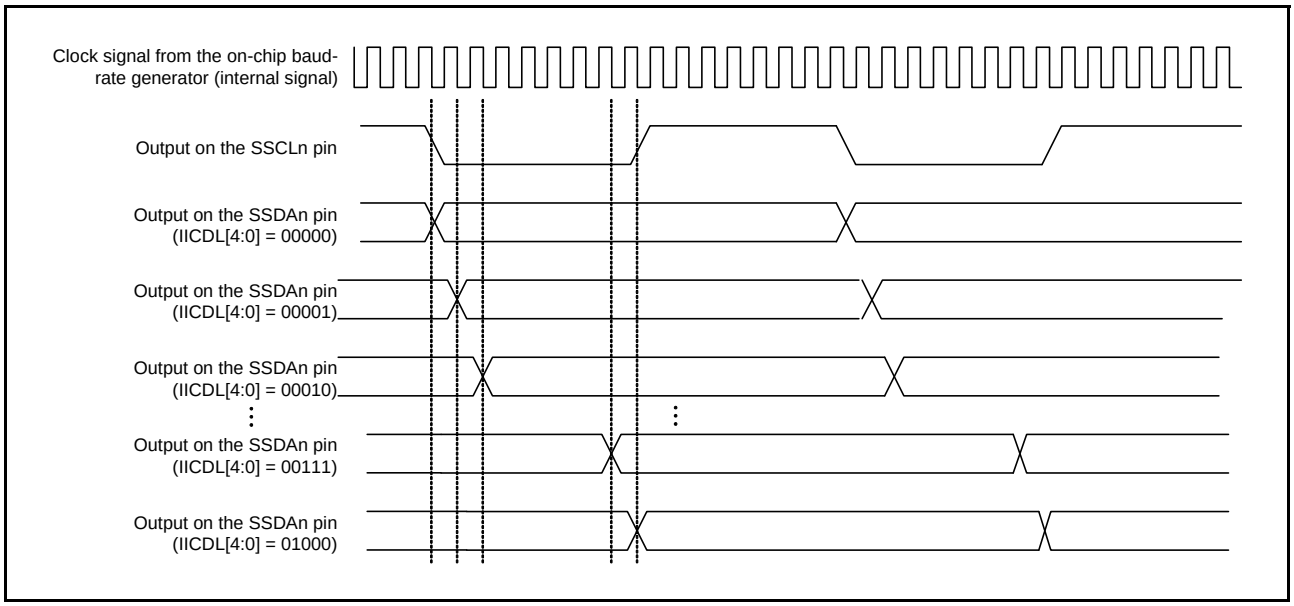
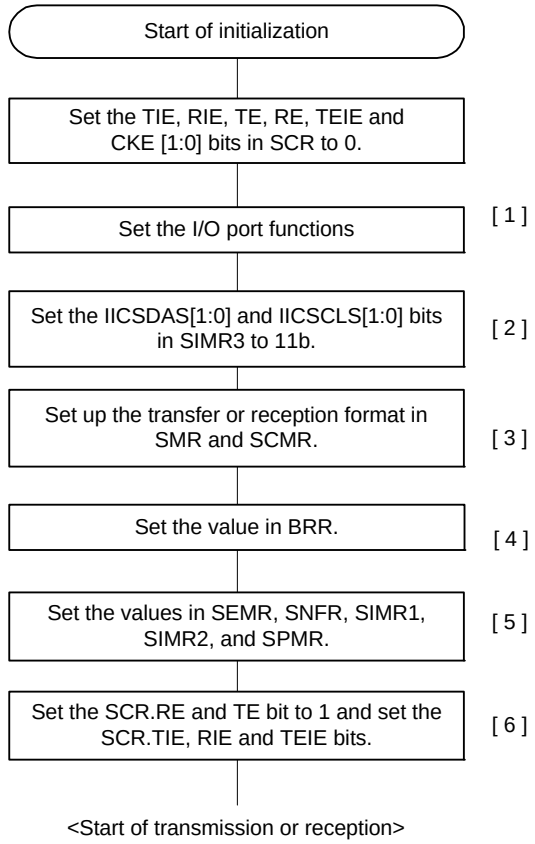


Figure 34.44 Timing of Delays in SSDA Output

34.7.4 SCI Initialization (Simple I²C Mode)

Before transferring data, write the initial value (00h) to SCR and initialize the interface in accordance with the flowchart shown as Figure 34.45.
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] Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR.
In SEMR, set the NFEN bit.
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, clear all the bits to 0.

[6] 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, clear the RIE bit). 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 34.45 Example of the Flow of SCI Initialization (for Simple I²C Mode)

34.7.5 Operation in Master Transmission (Simple I2C Mode)

Figure 34.46 and Figure 34.47 show examples of operations in master transmission and Figure 34.48 is a flowchart showing the procedure for data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (transmission and reception interrupts are in use) and the value of the SCR.RIE bit is assumed to be 0 (disabling reception interrupt requests). See Table 34.33 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 34.48 are repeated twice.
In simple I2C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock-synchronous transmission.

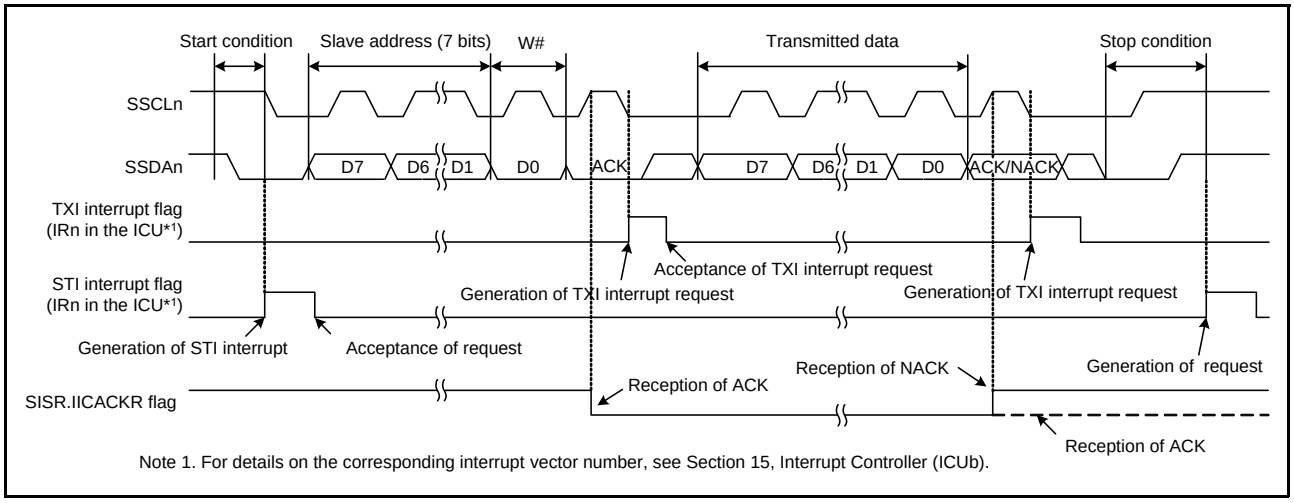


Figure 34.46 Example of Operations for Master Transmission in I2C Bus Mode (with Seven-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

When the SIMR2.IICINTM bit is set to 0 (ACK and NACK interrupts are used) 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 re-transmission, is performed by the NACK interrupt as the trigger.

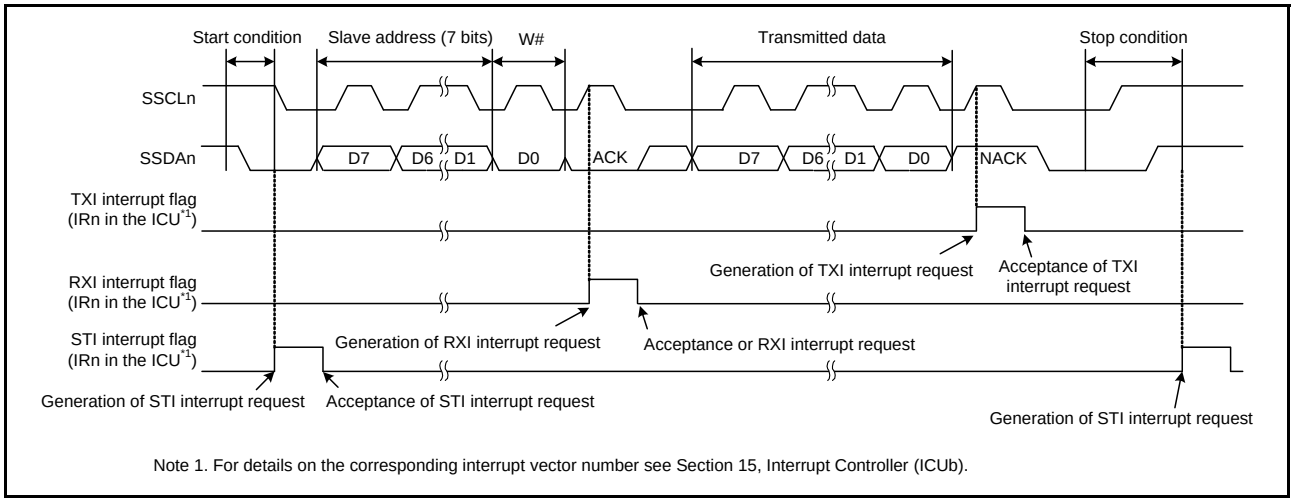


Figure 34.47 Example 2 of Operations for Master Transmission in Simple I2C Bus Mode (with Seven-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)

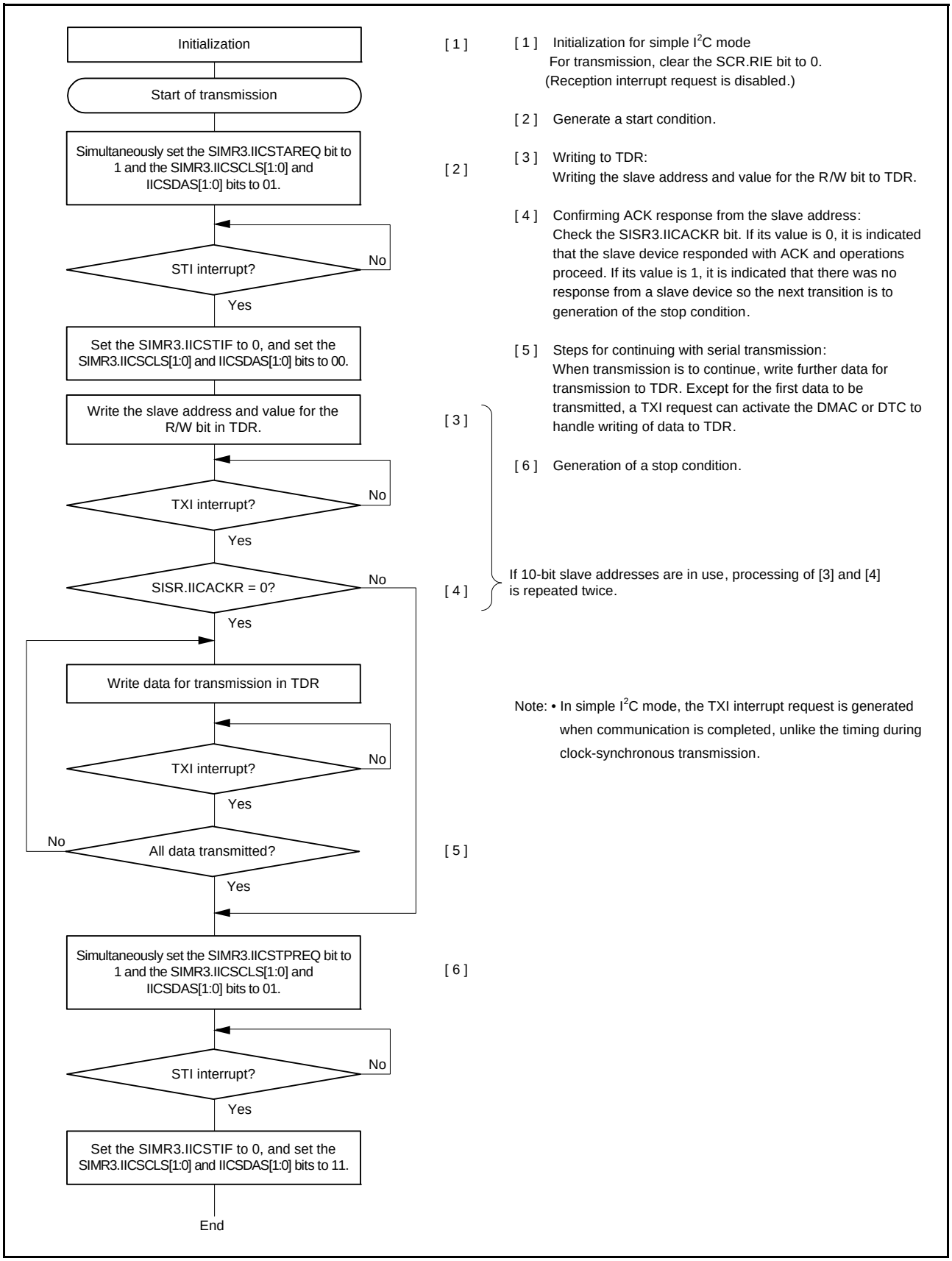
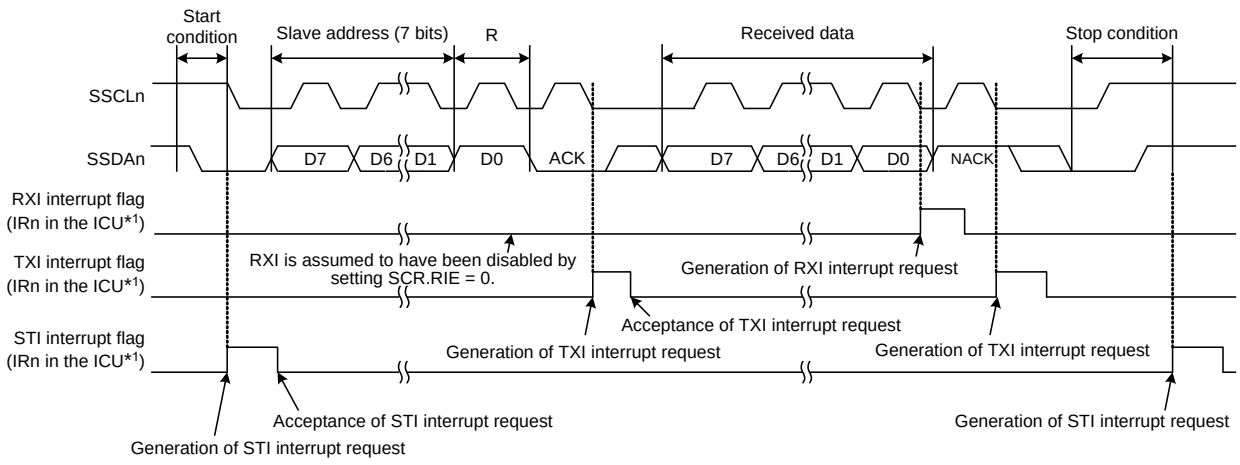


Figure 34.48 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

34.7.6 Master Reception (Simple I²C Mode)

Figure 34.49 shows an example of operations in simple I²C mode master reception and Figure 34.50 is a flowchart showing the procedure for master reception.
The value of the SIMR2.IICINTM bit is assumed to be 1 (transmission and reception interrupts are in use).
In simple I²C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock-synchronous transmission.



Note 1. For details on the corresponding interrupt vector number, see Section 15, Interrupt Controller (ICUb).

Figure 34.49 Example of Operations for Master Reception in I²C Bus Mode (with Seven-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

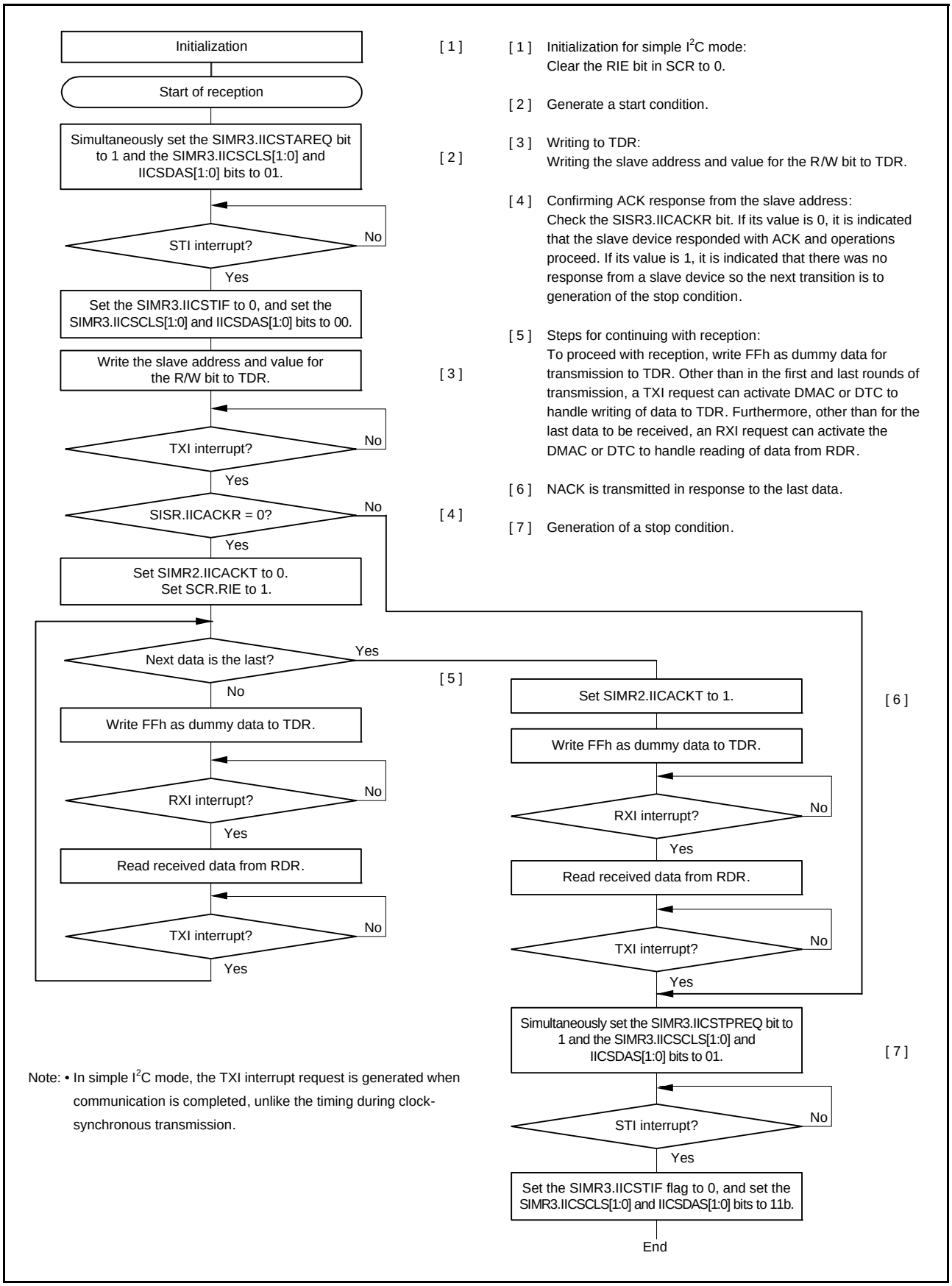


Figure 34.50 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

34.8 Operation in Simple SPI Mode

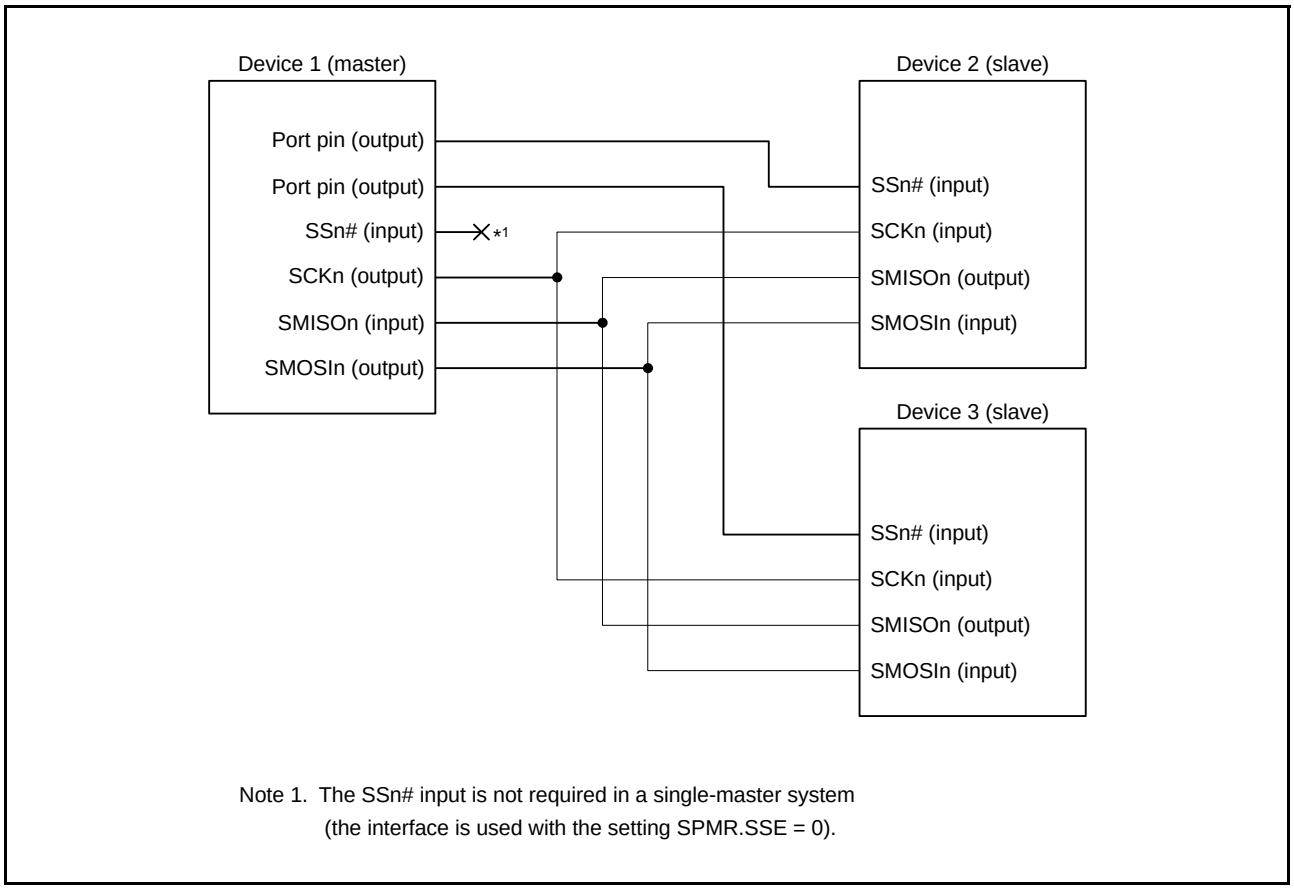
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock-synchronous mode (SCMR.SMIF = 0, SIMR.IICM = 0, SMR.CM = 1) plus setting the SSE bit in the SECR 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 34.51 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 eight 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 reception and transmission sections are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both sections have a buffered structure, writing of further data for transmission 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 34.51 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

34.8.1 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] = 00 or 01 and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10 or 11 and SPMR.MSS = 1). Table 34.29 lists the states of pins according to the mode and the level on the SSn# pin.

Table 34.29 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 transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when transmission is disabled (SCR.TE and RE bits = 00) in a multi-master configuration (SPMR.SSE = 1).

34.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00 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 output pins 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.

34.8.3 SS Function in Slave Mode

Setting the CKE[1:0] bits in the SCR to 10 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 SSn# pin input changes from low to high during transmission or reception, the operation is continued at the clock input through the SCKn pin until the transmission or reception is completed. Input of the clock signal from the SCKn pin is ignored until the input on the SSn# pin subsequently changes from the high to the low level.

34.8.4 Relationship between Clock and Transmit/Receive Data

The SPMR.CKPOL and CKPH bits 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 34.52. The relation is the same for both master and slave operation. This is the same as when the level on the SSn# pin is high. The SSn# pin can be used for another purpose. For details, see section 34.8.2, SS Function in Master Mode.

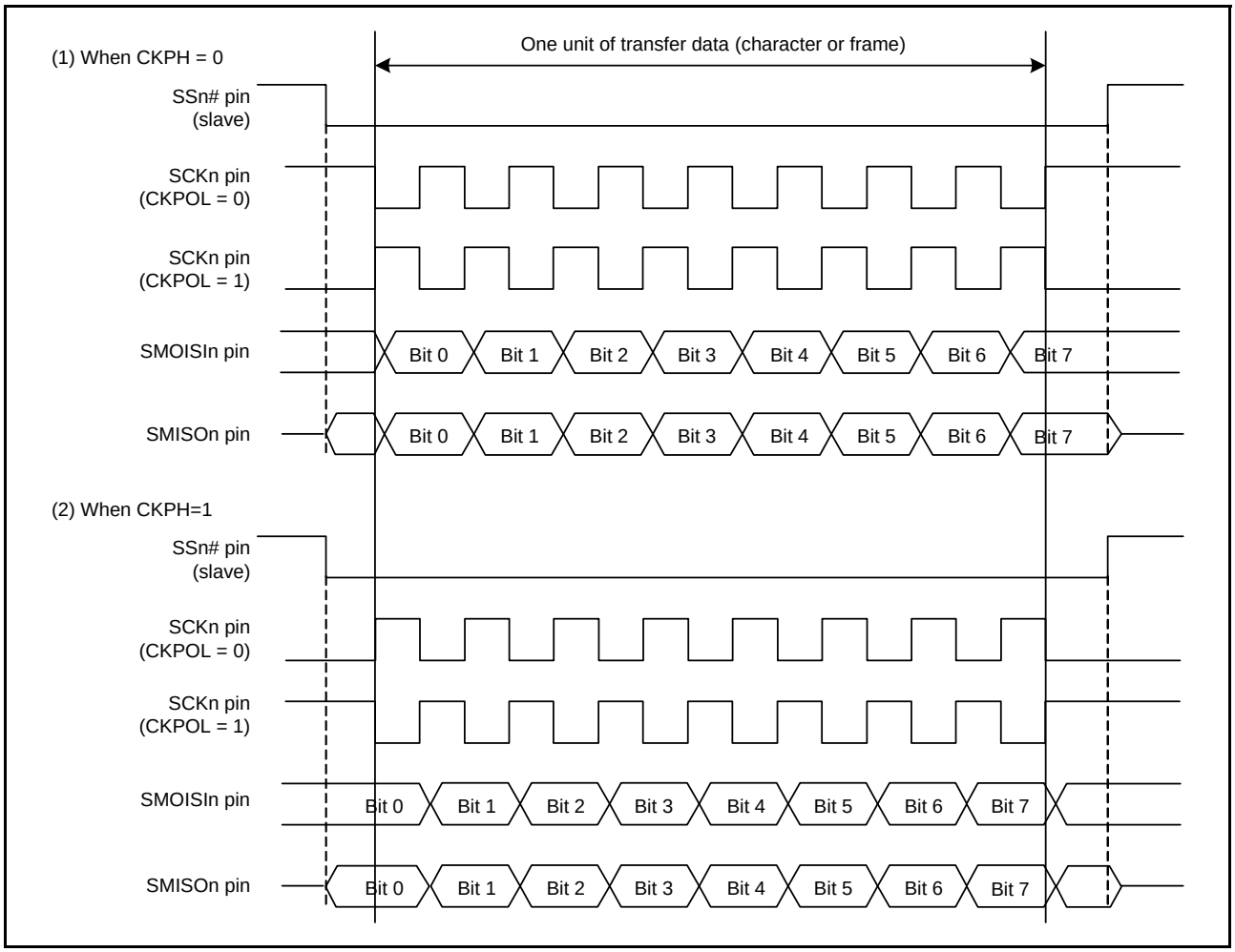


Figure 34.52 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

34.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock-synchronous mode (Figure 34.21, Sample SCI Initialization Flowchart). The CKPOL and CKPH bits in the SPMR must be set to endure 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.

Note that the ORER, FER, and PER flags, as well as the RDR, are not initialized even when the RE bit is set to 0.

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 transmission interrupt (TXI) if the value of the TIE bit in the SCR is 1 at the time.

34.8.6 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.

34.9 Extended Serial Mode Control Section: Description of Operation

34.9.1 Serial Transfer Protocol

In conjunction with an SCIc module, the extended serial mode control section of an SCId module can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 34.53.
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.

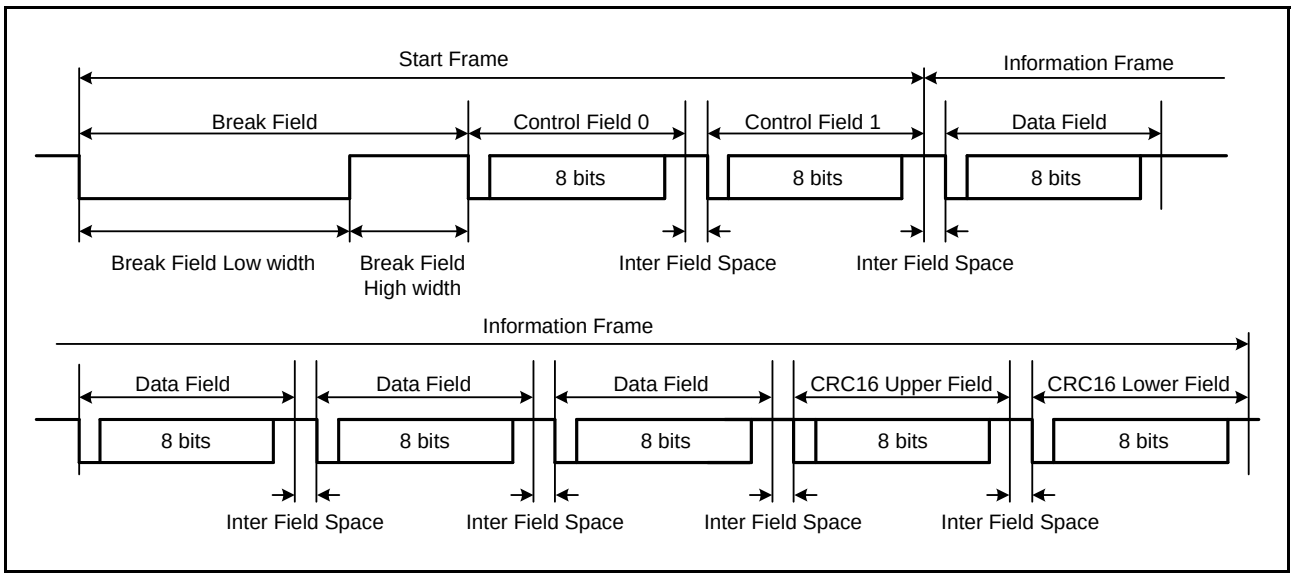


Figure 34.53 Protocol for Serial Transfer by the Extended Serial Mode Control Section

34.9.2 Transmitting a Start Frame

Figure 34.54 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 34.55 and Figure 34.56 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 TCST bit in TCR starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to the TCNT and TPRE settings.
(2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.
(3) Writing 0 to the TCST bit in TCR stops counting by the timer, and SCI2 is used to send the data for Control Field 0. After the Break Field low width, stop counting before the next underflow occurs.
(4) Once the data for Control Field 0 have been transmitted, SCI12 is used to send the data for Control Field 1.
(5) Once the data for Control Field 1 have been transmitted, SCI12 is used to send an Information Frame.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

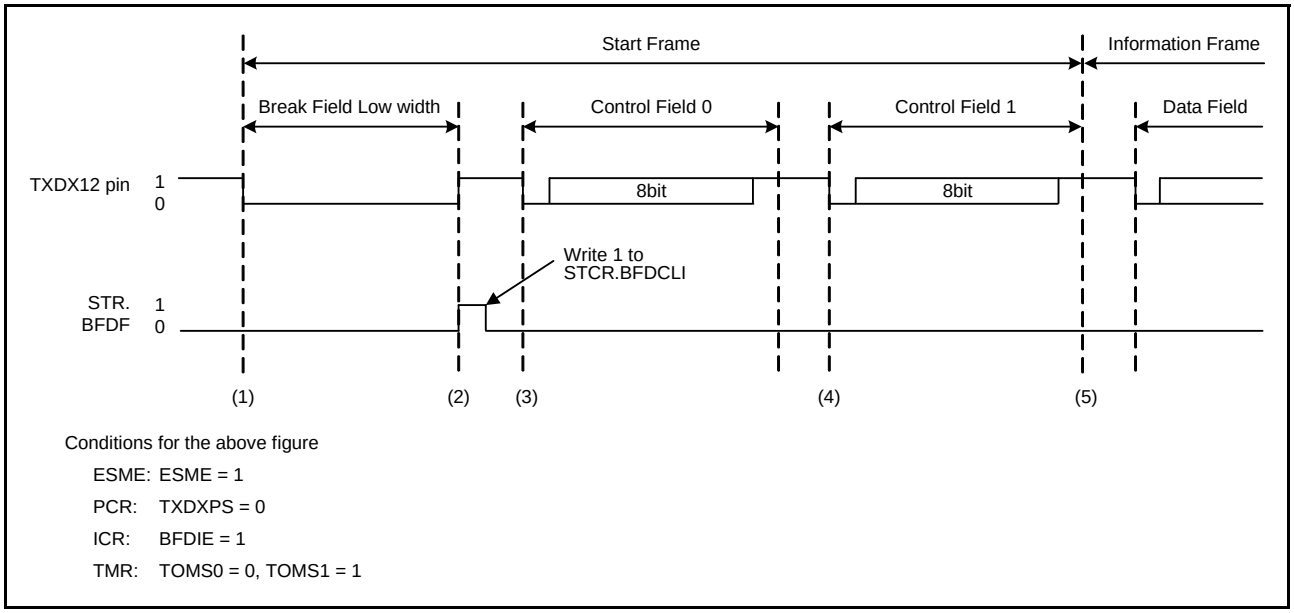


Figure 34.54 Example of Operations at the Time of Start-Frame Transmission

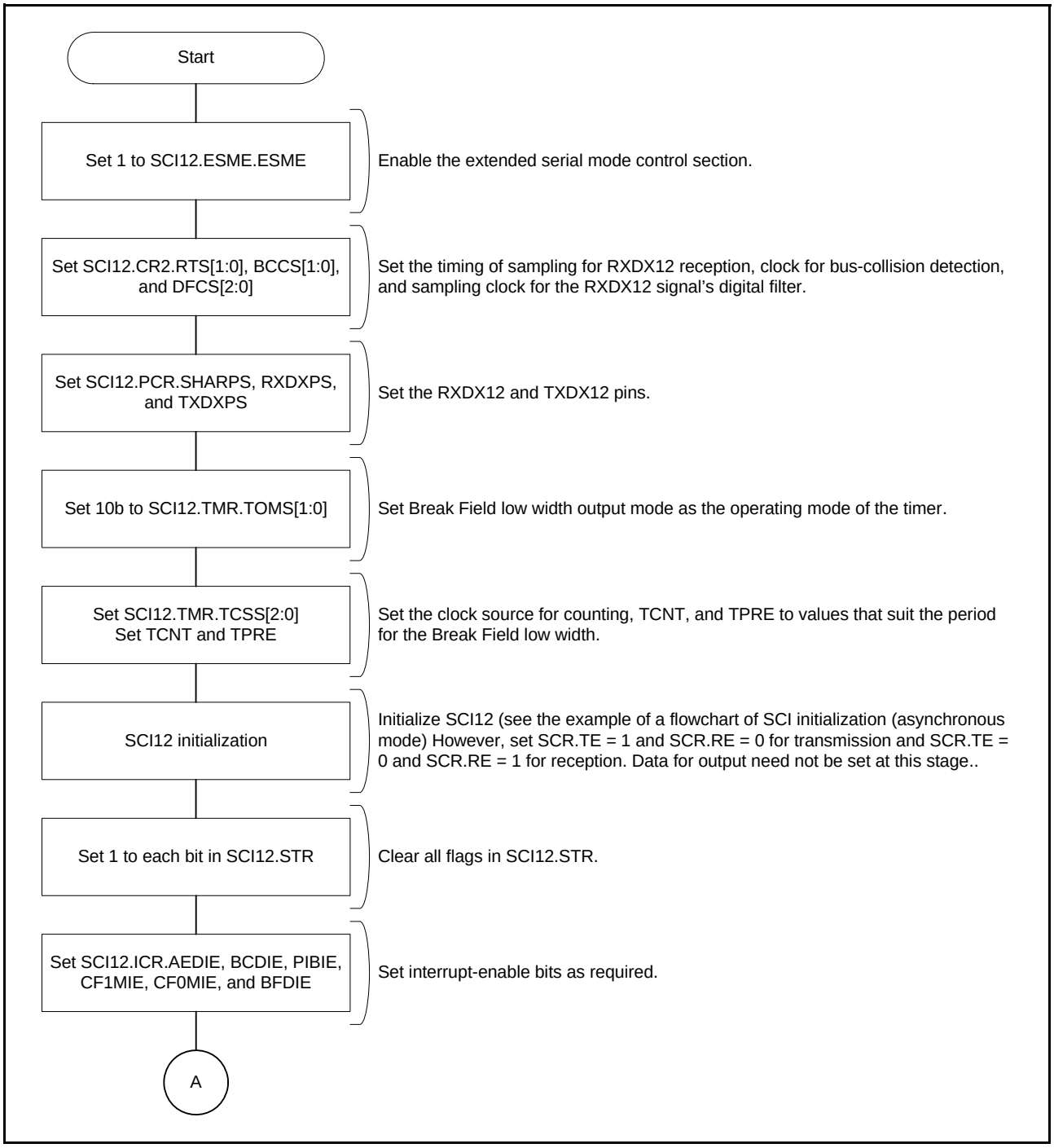


Figure 34.55 Sample Flowchart for Transmission of a Start Frame (1)

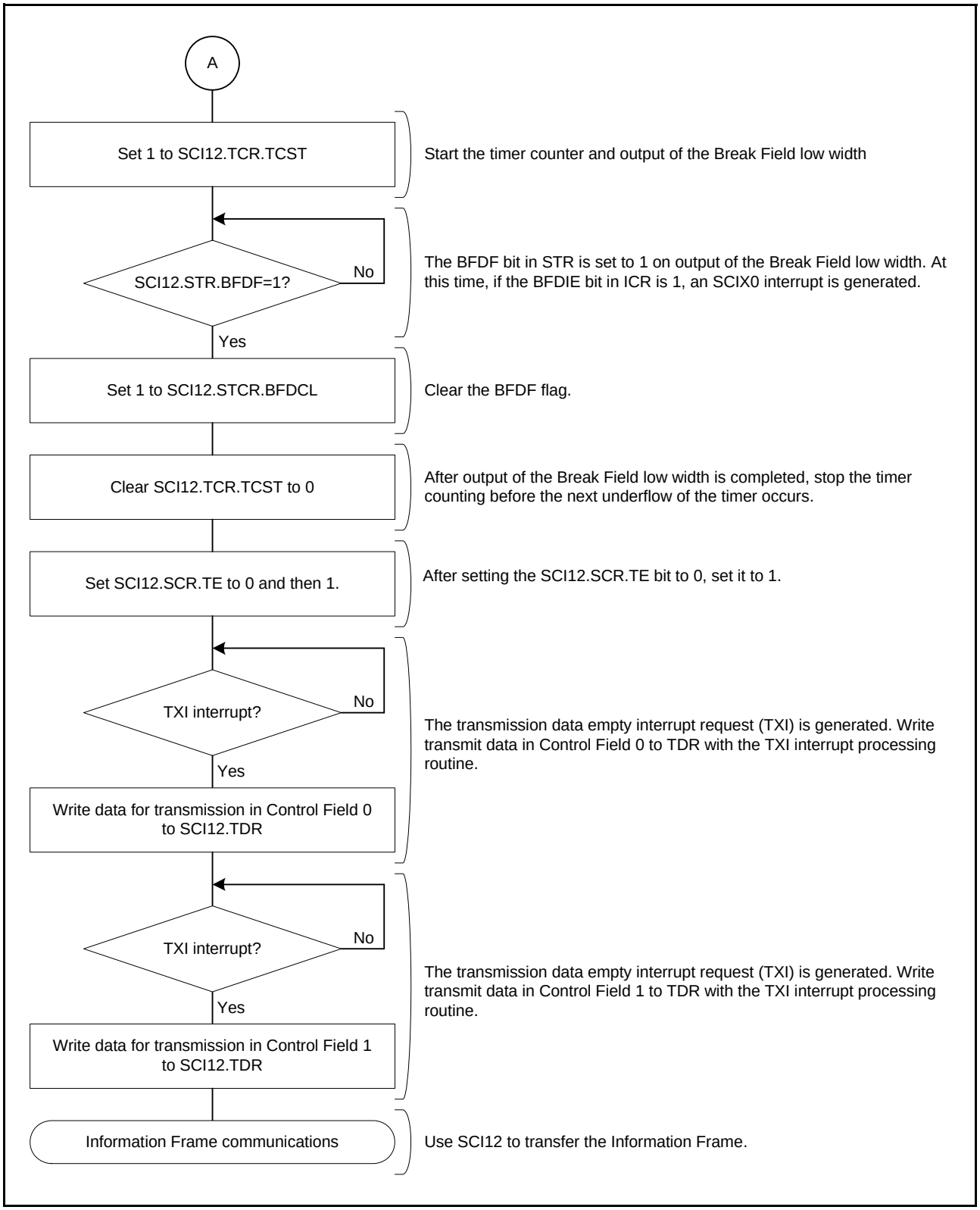


Figure 34.56 Sample Flowchart for Transmission of a Start Frame (2)

34.9.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 34.30.

Table 34.30 Structures of Start Frames

Bit Setting		Structures of Start Frames
BFE	CF0RE	
0	0	Control Field1 (8 bits) → Information Frame
0	1	Control Field0 (8 bits) → Control Field1 (8 bits) → Information Frame
1	0	Break Field (Low width) → Control Field1 (8 bits) → Information Frame
1	1	Break Field (Low width) → Control Field0 (8 bits) → Control Field1 (8 bits) → Information Frame

Figure 34.57 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 34.58 and Figure 34.59 are flowcharts for the reception of a Start Frame, and Figure 34.59 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 SDST bit in CR3 enables detection of the Break Field low width. RXDX12 input to the SCI12 module 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 TCNT and TPRE is detected as the Break Field low width. At this time, the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.
(3) When the input from the RXDX12 pin goes high after the Break Field low width, the RXDSF bit in CR0 becomes zero and reception of Control Field 0 by the SCI12 module starts.
(4) If the data received in Control Field 0 match the data set in CF0DR, the CF0MF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the CF0MIE bit in ICR is 1. Reception of Control Field 1 by the SCI12 module starts after that. If the data received in Control Field 0 do not match the data set in CF0DR, 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 PCF1DR and SCF1DR, the CF1MF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the CF1MIE bit in ICR is 1. Transfer of the Information Frame by the SCI12 module starts after that. If the data received in Control Field 1 do not match the data set in either or both of 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.

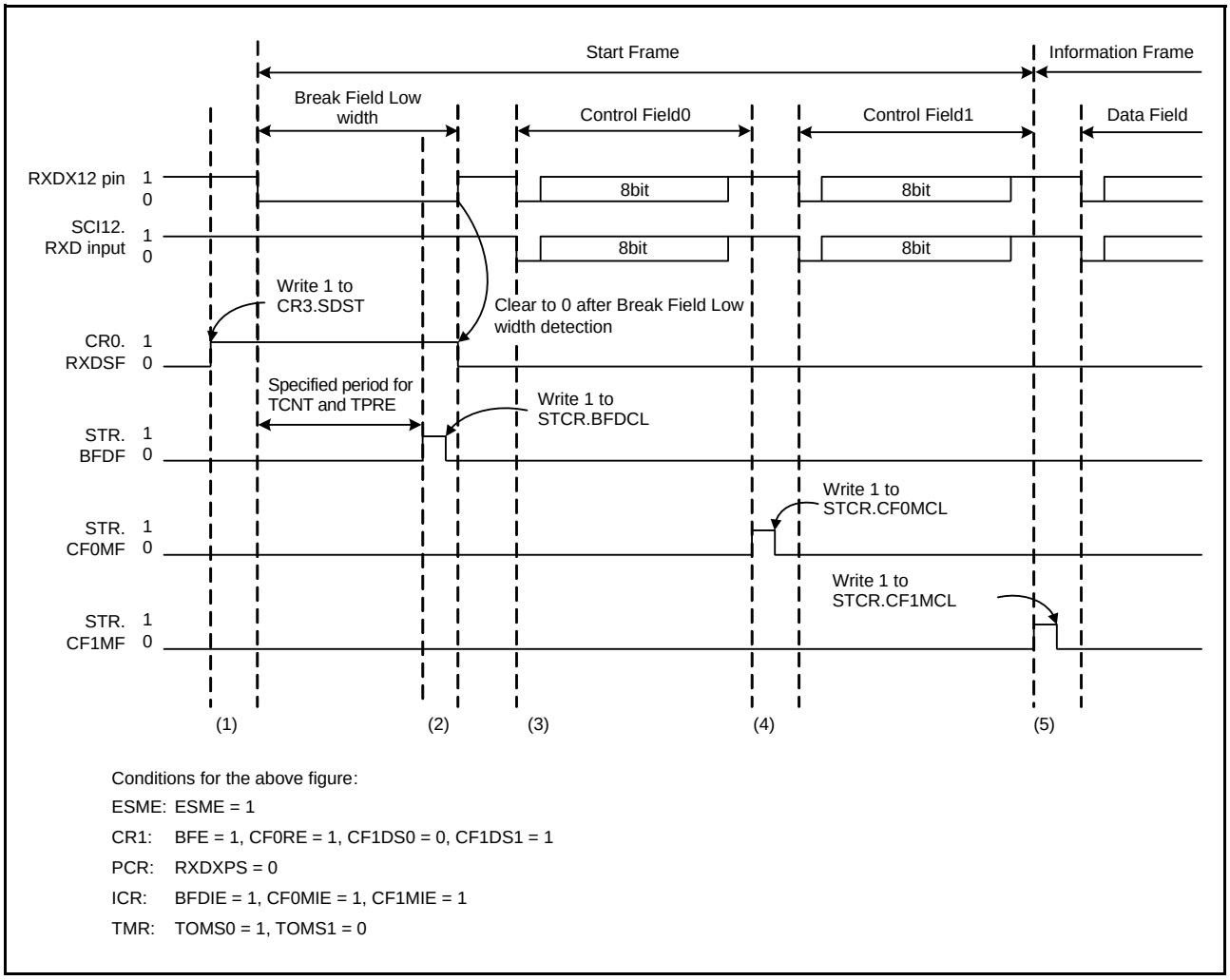


Figure 34.57 Example of Operations at the Time of Start-Frame Reception

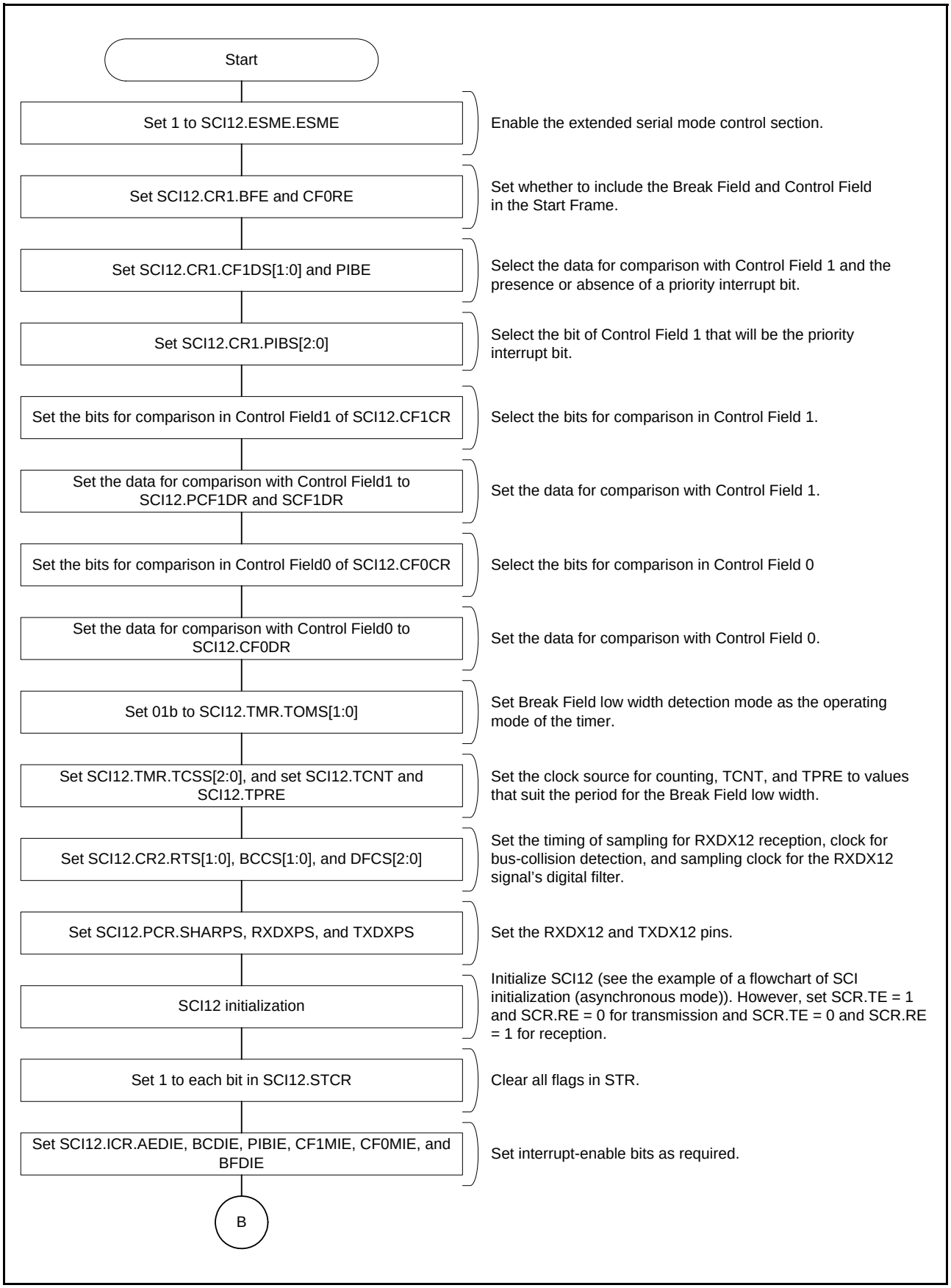


Figure 34.58 Sample Flowchart for Reception of a Start Frame (1)

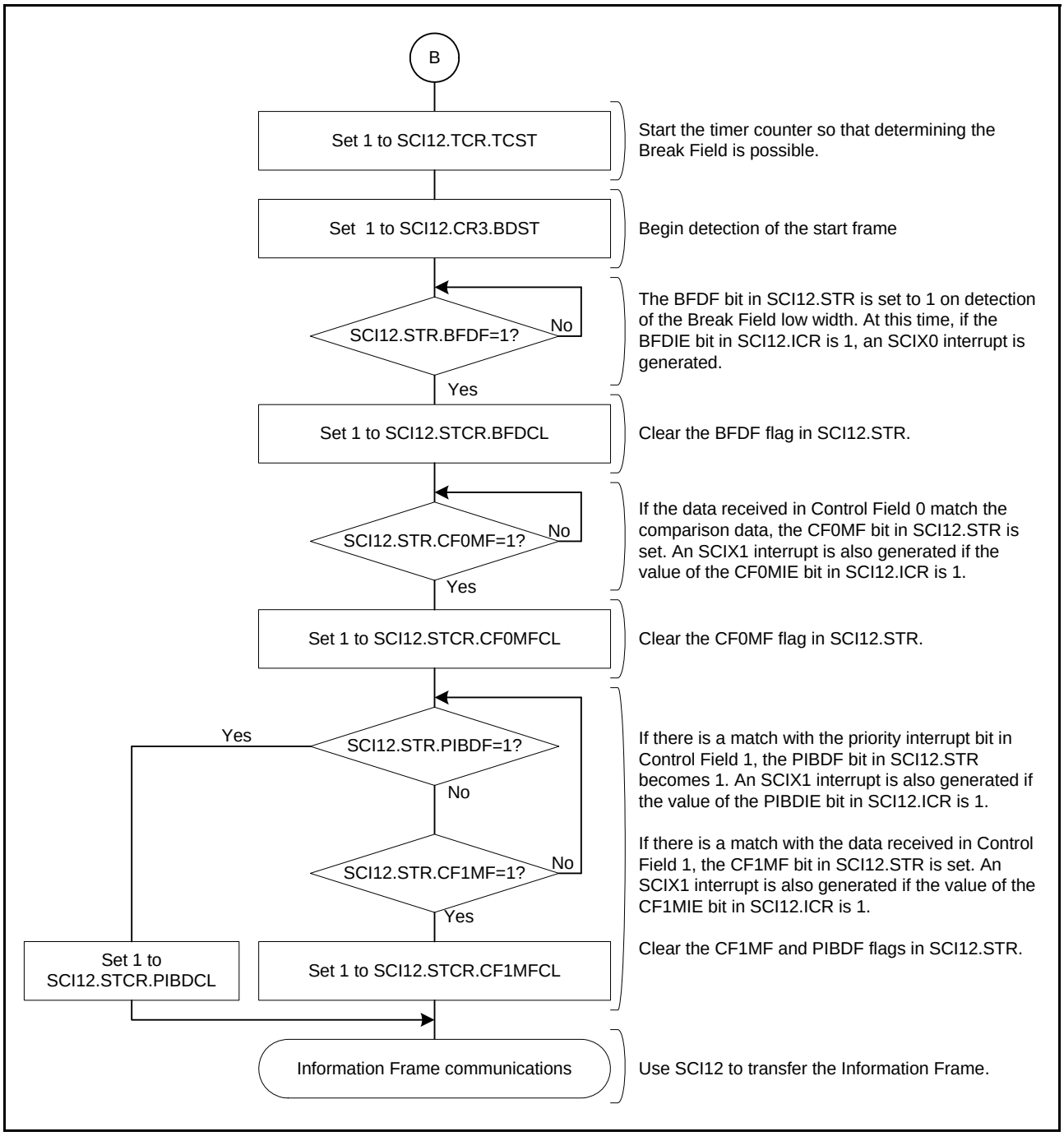


Figure 34.59 Sample Flowchart for Reception of a Start Frame (2)

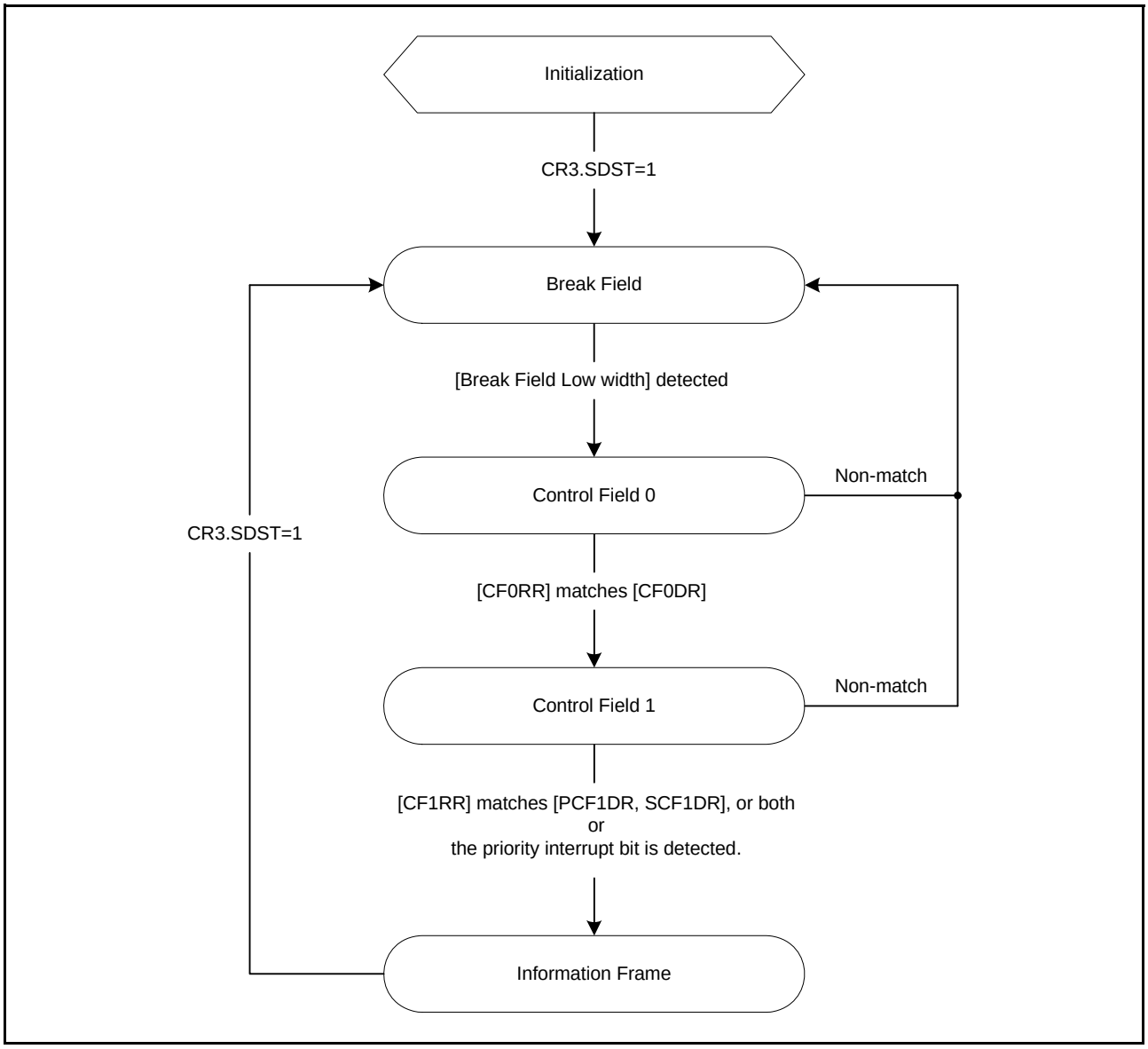


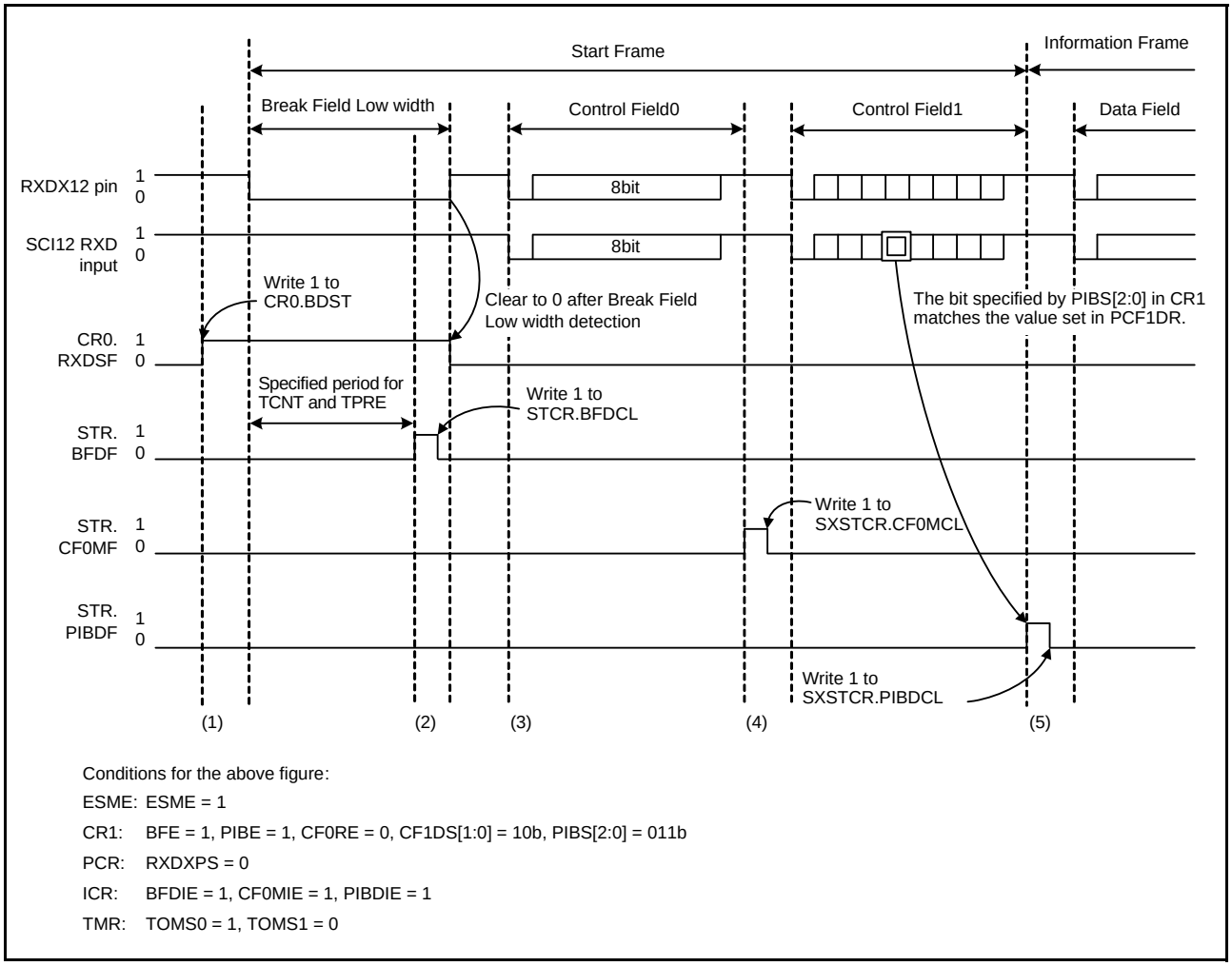
Figure 34.60 Diagram of State Transitions at the Time of Start-Frame Reception

34.9.3.1 Priority Interrupt Bit

Figure 34.61 shows an example of operation in Start-Frame reception where a priority interrupt bit is in use. Setting the PIBE bit in CR1 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 34.57, for Start-Frame reception.

(5) If the value of the bit selected by the PIBS[2:0] bits in CR1 matches the corresponding bit in PCF1DR, the PIBDF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the PIBDIE bit in ICR is 1. Transfer of the Information Frame by the SCI12 module starts after that. If the data received in Control Field 1 do not match the data set in either or both of PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.



Conditions for the above figure:
ESME: 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: TOMS0 = 1, TOMS1 = 0

Figure 34.61 Sample Flow Chart for Reception of a Start Frame (with Priority Interrupts in Use)

34.9.4 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 module are in progress when the ESMER.ESME and the SCI12.SCI. TE are set to 1.

Figure 34.62 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 CR2.BCCS[1:0] as the sampling clock, and the BCDF bit in STR 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 BCDIE bit in ICR is 1.

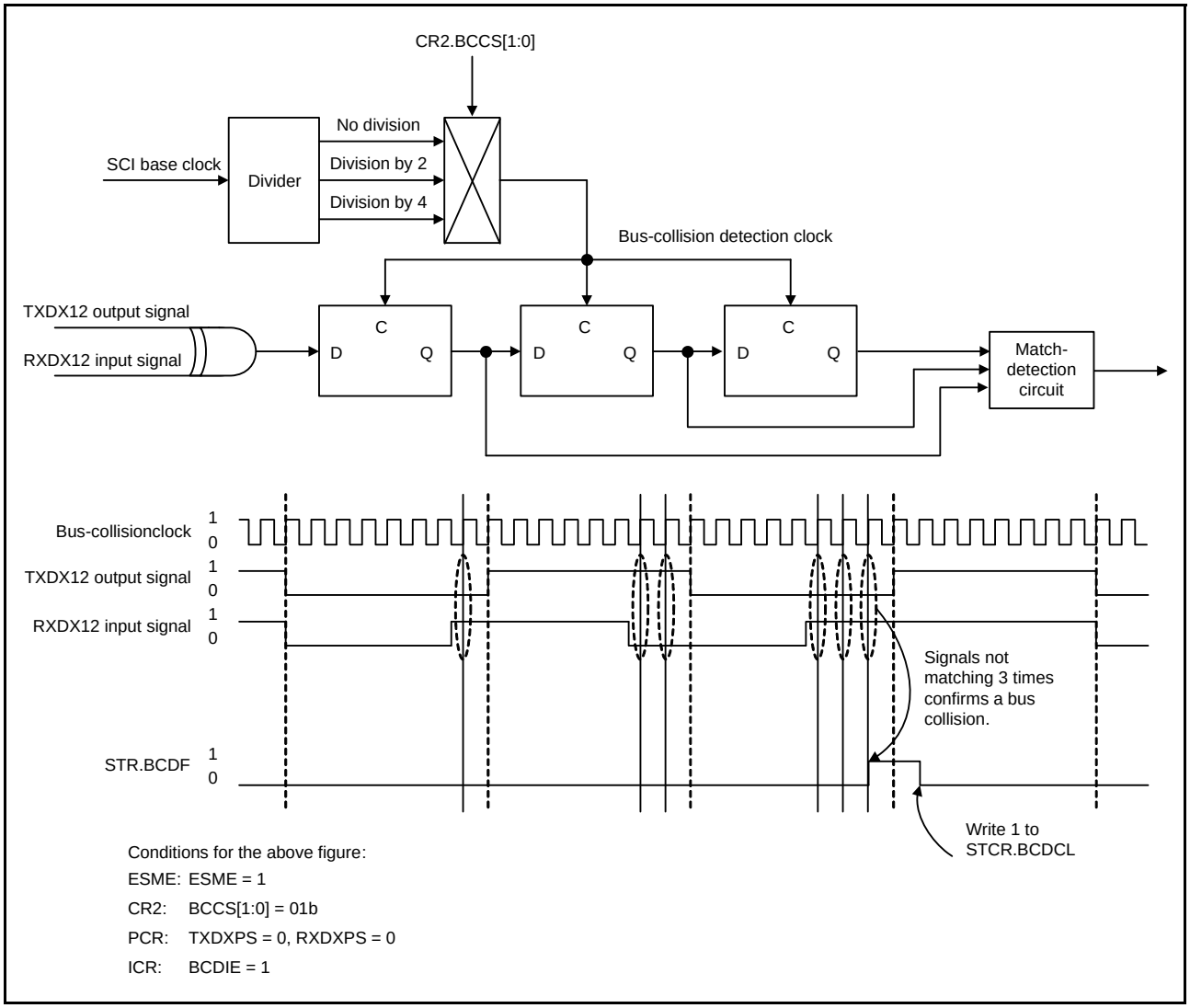


Figure 34.62 Example of Operations with Bus-Collision Detection

34.9.5 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 DFCS[2:0] bits in CR2 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 34.63 shows an example of operations with the digital filter.

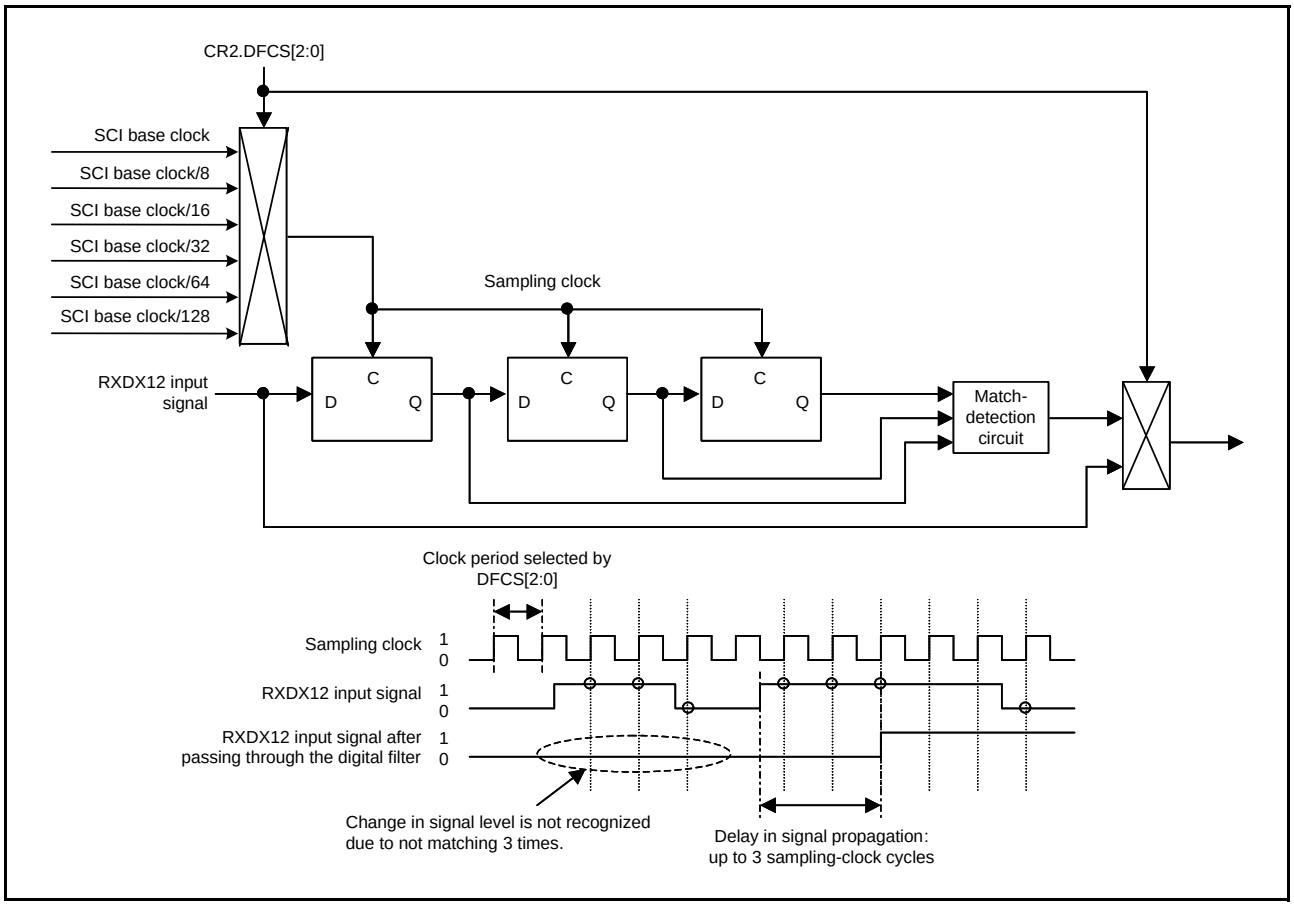
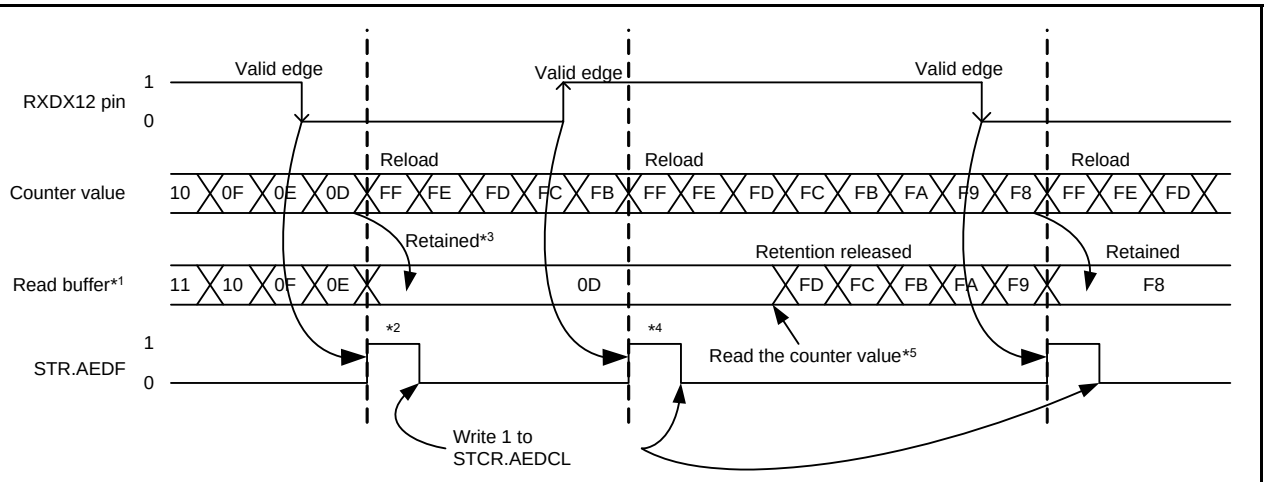


Figure 34.63 Example of Operations with the Digital Filter

34.9.6 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 34.64 shows an example of operations for bit-rate measurement.

(1) Writing 1 to the BRME bit in CR0 enables bit-rate measurement. Only set BRME to 1 when you wish to proceed with bit-rate measurement. Furthermore, bit-rate measurement will not proceed during a Break Field, even if BRME 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 AEDIE bit in ICR is 1. Retention by 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 module. To disable the bit-rate measurement after a match with Control Field 1, write 0 to the BRME bit in CR0.



Note 1. The contents of the read buffers are read by reading TCNT and TPRE.
Note 2. The AEDF bit in STR becomes 1 on detection of a valid edge.
Note 3. The current counter value is retained in the read buffers on detection of a valid edge.
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 TCNT or TPRE.

Figure 34.64 Example of Operations for Bit-Rate Measurement

34.9.7 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 module by setting the RS0 and RS1 bits in CR2 to select the rising edges of 8, 10, 12, or 14 cycles of SCI base clock. If the value of the ABCS bit in SEMR is 1, the bits select the rising edges of 4, 5, 6, or 7 cycles of the SCI base clock of the SCI12 module. Figure 34.65 shows timing for the sampling of data received through RXDX12.

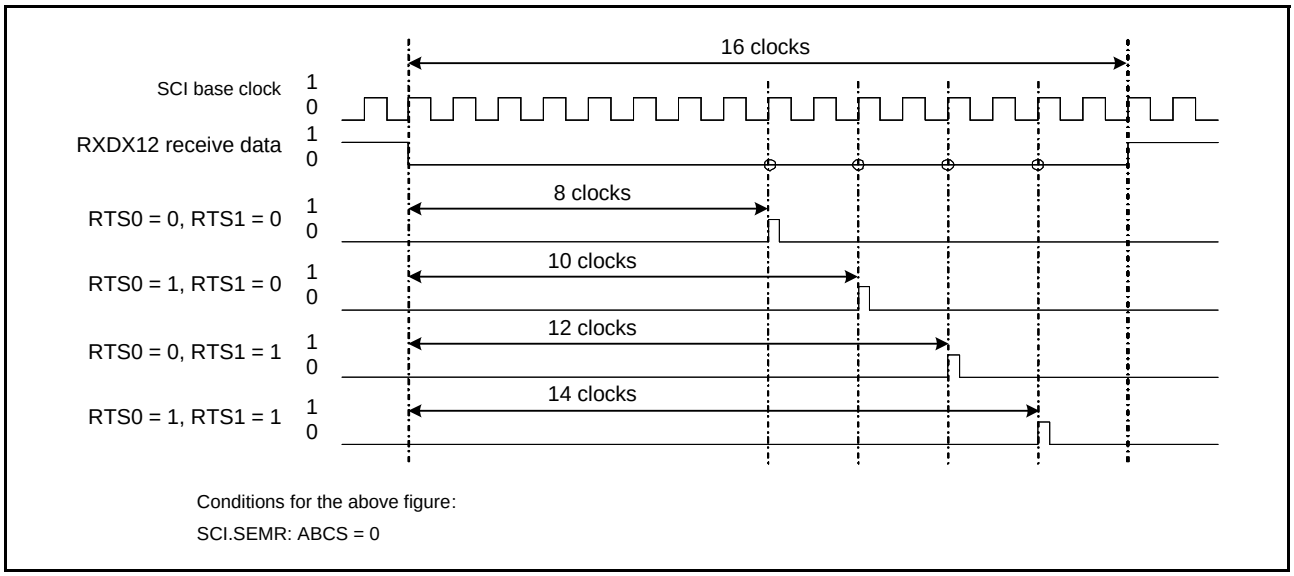


Figure 34.65 Timing for Sampling of Data Received through RXDX12

34.9.8 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 time of transmitting a Start Frame. Setting TOMS0 to 0 and TOMS1 to 1 in TMR switches operation to Break Field low width output mode. The TCSS[2:0] bits in TMR select the clock source for the counter. When the TCST bit in TCR 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 BFDF bit in the STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1. When 0 is written to the TCST bit in TCR, counting stops after reloading of TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 34.66 shows an example of operations in Break Field low width output mode.

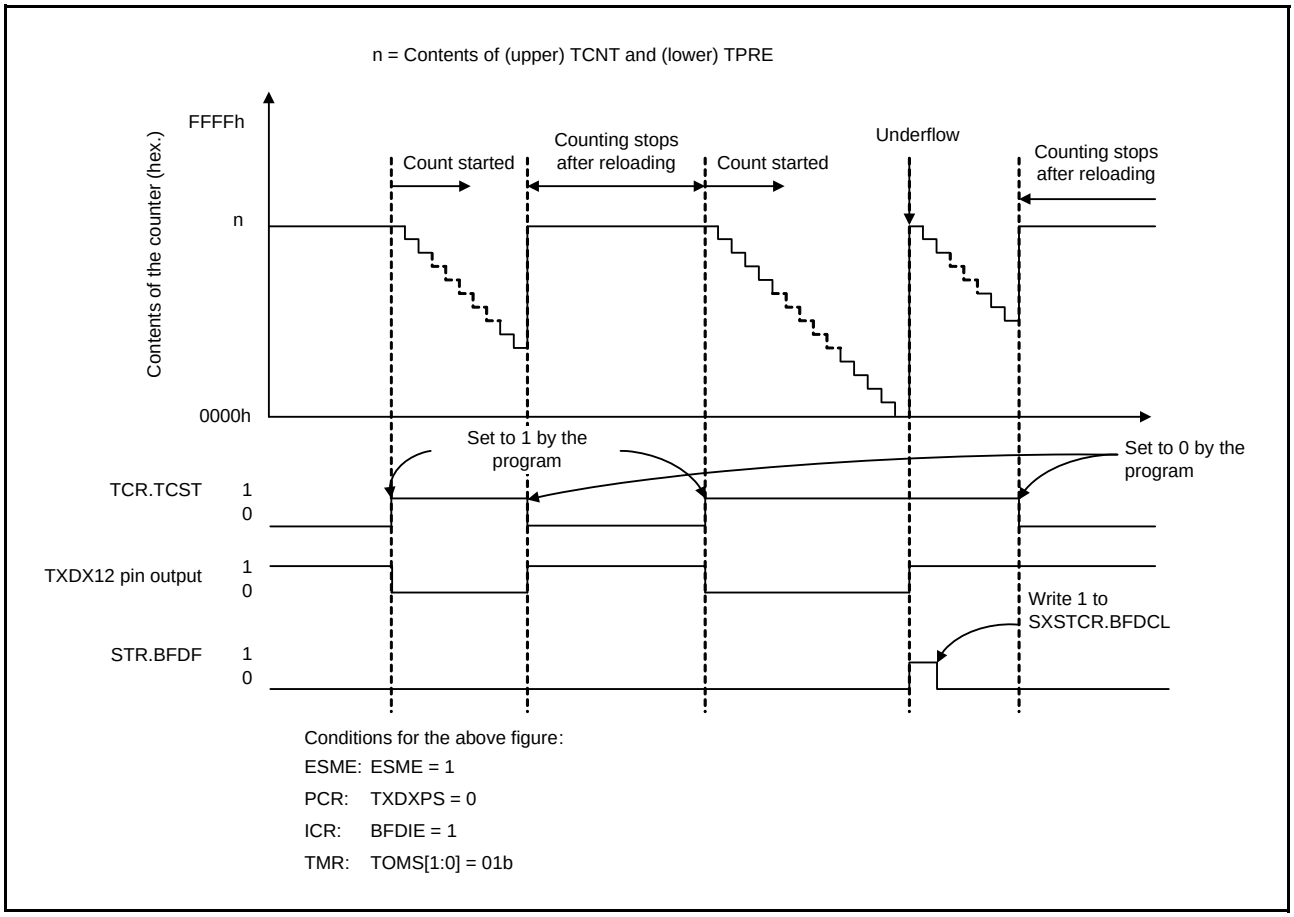


Figure 34.66 Example of Operations in Break Field Low Width Output Mode

(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 time of receiving a Start Frame. Setting TOMS0 to 1 and TOMS1 to 0 in TMR switches operation to Break Field low width determination mode. The TCSS[2:0] bits in TMR select the clock source for the counter. When the TCST bit in TCR 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, 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 BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR 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 34.67 shows an example of operations in Break Field low width output mode.

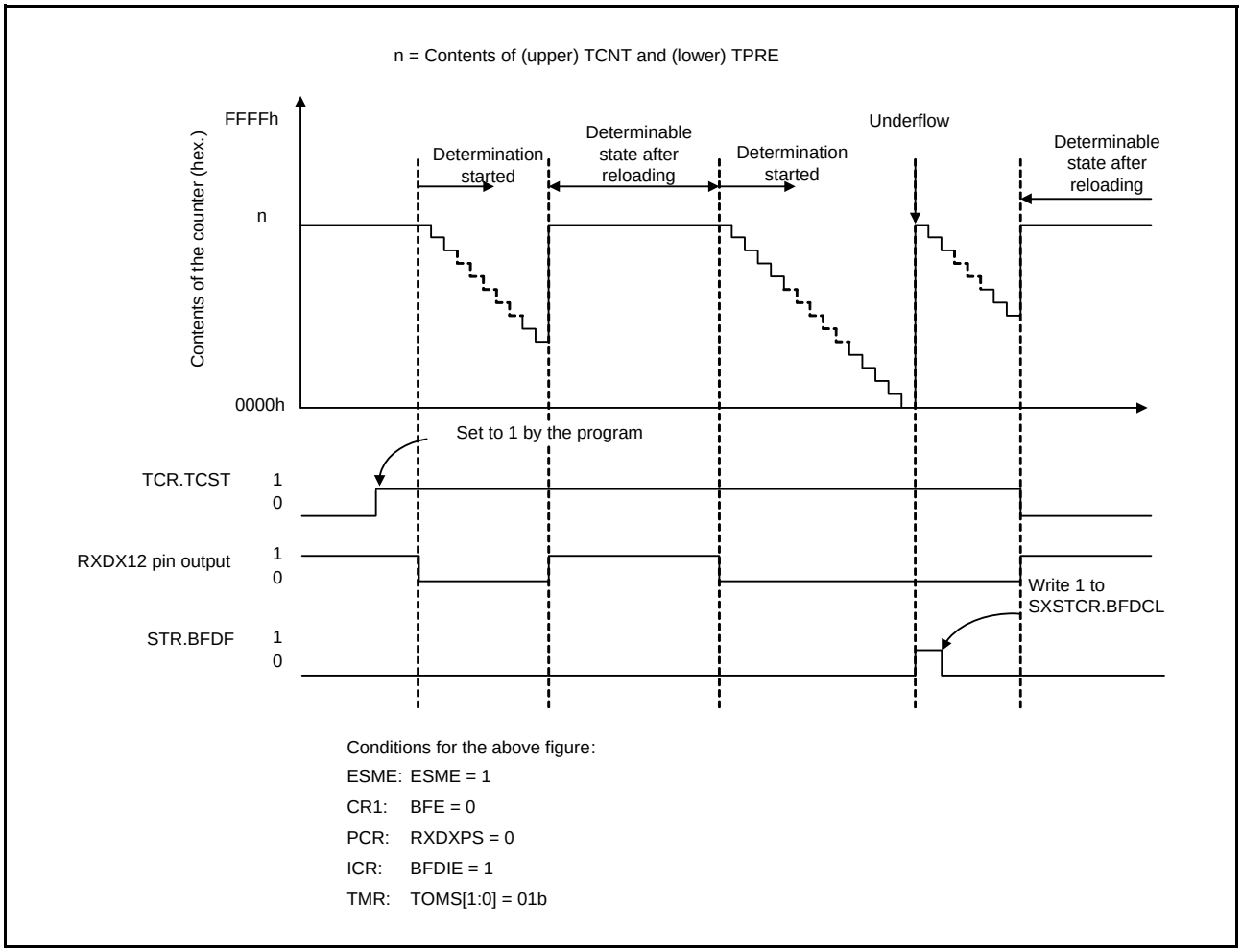


Figure 34.67 Example of Operations in Break Field Low Width Determination Mode

(3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting TOMS0 to 0 and TOMS1 to 0 in TMR switches operation to timer mode. The TCSS[2:0] bits in TMR select the clock source for counting. Counting starts when 1 is written to the TCST bit in TCR and stops when 0 is written to TCST. TPRE and TCNT both count down. TPRE counts cycles of the clock source for counting, and underflows of TPRE provide the clock source for counting by TCNT. When the timer underflows, the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.

34.10 Noise Cancellation Function

Figure 34.68 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 I^2C 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.

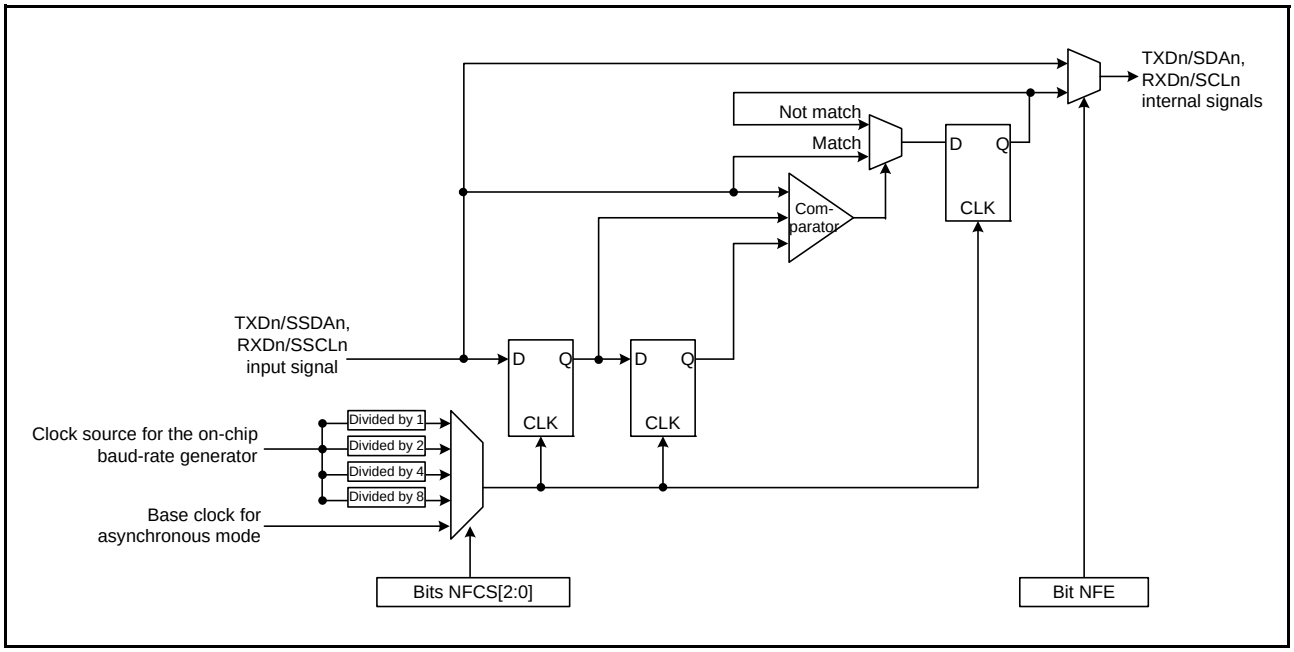


Figure 34.68 Block Diagram of Digital Noise Filter Circuit

34.11 Interrupt Sources

34.11.1 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 interrupt controller 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 interrupt controller 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.

34.11.2 Interrupts in Serial Communications Interface and Simple SPI Mode

Table 34.31 lists interrupt sources in serial communication interface 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 SCR.
If the SCR.TIE bit is 1, a TXI interrupt request is generated when data for transmission are transferred from the TDR 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.*1
When new data are not written by the time of transmission of the last bit of the current data for transmission 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 data for transmission are written to the TDR, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.
Writing data to the TDR 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 are 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. To temporarily prohibit 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 prohibiting and permitting 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 34.31 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	—	Possible	Possible	↑
TXI	Transmit data empty	—	Possible	Possible	
TEI	Transmit end	TEND	Not possible	Not possible	Low

34.11.3 Interrupts in Smart Card Interface Mode

Table 34.32 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 34.32 SCI Interrupt Sources

Name	**Interrupt Source**	Interrupt Flag	DTC Activation	DMACA 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 SSR 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 cleared to 0 when the DTC or DMAC transfers the data.
If an error occurs, the SCI automatically re-transmits 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 SSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in SCR 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, see section 17, DMA Controller (DMACA) and section 19, 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.

34.11.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 34.33. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.
The DTC and DMAC can also be used to handle transfer in simple I2C mode.
When the value of the IICINTM bit in SIMR2 is 1, an RXI request will be generated on the falling edge of the SSCL 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 data for transmission.
When the value of the IICINTM bit in SIMR2 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, IICSTAREQ, and IICSTPREQ bits in SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 34.33 SCI Interrupt Sources

Name	**Interrupt Source**	Interrupt Flag	DTC Activation	DMAC Activation	Priority
RXI	Reception, ACK detection	—	Possible	Possible	High
TXI	Transmission, NACK detection	—	Possible*1	Possible*1	↑
STI	Completion of generating 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 (selecting reception and interrupts).

34.11.5 Interrupts 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 34.34.

Table 34.34 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

34.12 Usage Notes

34.12.1 Setting the Module Stop Function

Module-stop control registers B and C (MSTPCRB and MSTPCRC) are used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. The registers of the modules only become accessible after release from the module-stop state. For details, refer to section 11, Low Power Consumption.

34.12.2 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 SSR is set to 1 (framing error), and the PER flag in SSR may also be set to 1 (parity error). The SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is cleared to 0 (no framing error), it will be set to 1 again.

34.12.3 The Mark State and Production of Breaks

When the SCR.TE bit is 0 (prohibiting serial transmission), 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 (permitting serial transmission), the I/O port function is used to set the TXDn pin to output 1 and thus place the transfer circuit in the mark state (state of having the value 1). On the one hand, if a break is to be output at the time of data transmission, the SCR.TE bit is cleared to 0 after the I/O port function settings are used to set up output of a 0 on the TXDn pin. When the SCR.TE bit is cleared to 0, the transmission section is initialized without connection to the current state of transmission, the TXDn pin becomes an I/O port pin, and the output from the TXDn pin changes to 0 or 1 in accord with the settings for the I/O port function.

34.12.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER) in SSR is set to 1, even if data is written to TDR. Be sure to clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0 (serial reception disabled).

34.12.5 Writing Data to TDR

Data can always be written to TDR. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. Be sure to write transmit data to TDR in the TXI interrupt request processing routine.

34.12.6 Restrictions on Clock Synchronous Transmission

When the external clock source is used as a synchronization clock, update TDR by the DMAC or DTC and wait for at least five clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR is updated, the SCI may malfunction.

34.12.7 Restrictions on Using DTC or DMAC

When using the DMAC or DTC to read RDR, be sure to set the receive end interrupt (RXI) as the activation source of the relevant SCI.

34.12.8 Points to Note on Starting Transfer

At the point where transfer starts when the interrupt status flag 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).

- 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 out the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has actually become 0.
- Set the interrupt status flag in the interrupt controller to 0.

34.12.9 SCI Operations during Low Power Consumption State

(1) Transmission

Before specifying the module stop state or making a transition to software standby mode, stop the transmit operations (TIE = TE = TEIE = 0 in SCR). TSR, TDR, and SSR are reset by clearing the TE bit. The states of the output pins in the module stop state or in software standby mode depend on the port settings, and the output pins are held high after cancellation. If the transition is made during data transmission, the data being transmitted will be undefined.
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 34.69 shows a sample flowchart for transition to software standby mode during transmission. Figure 34.70 and Figure 34.71 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 transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE bit to 1. The TXI interrupt flag is set to 1 and transmission starts using the DTC.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (RE = 0 in SCR). 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 34.72 shows a sample flowchart for transition to software standby mode during reception.

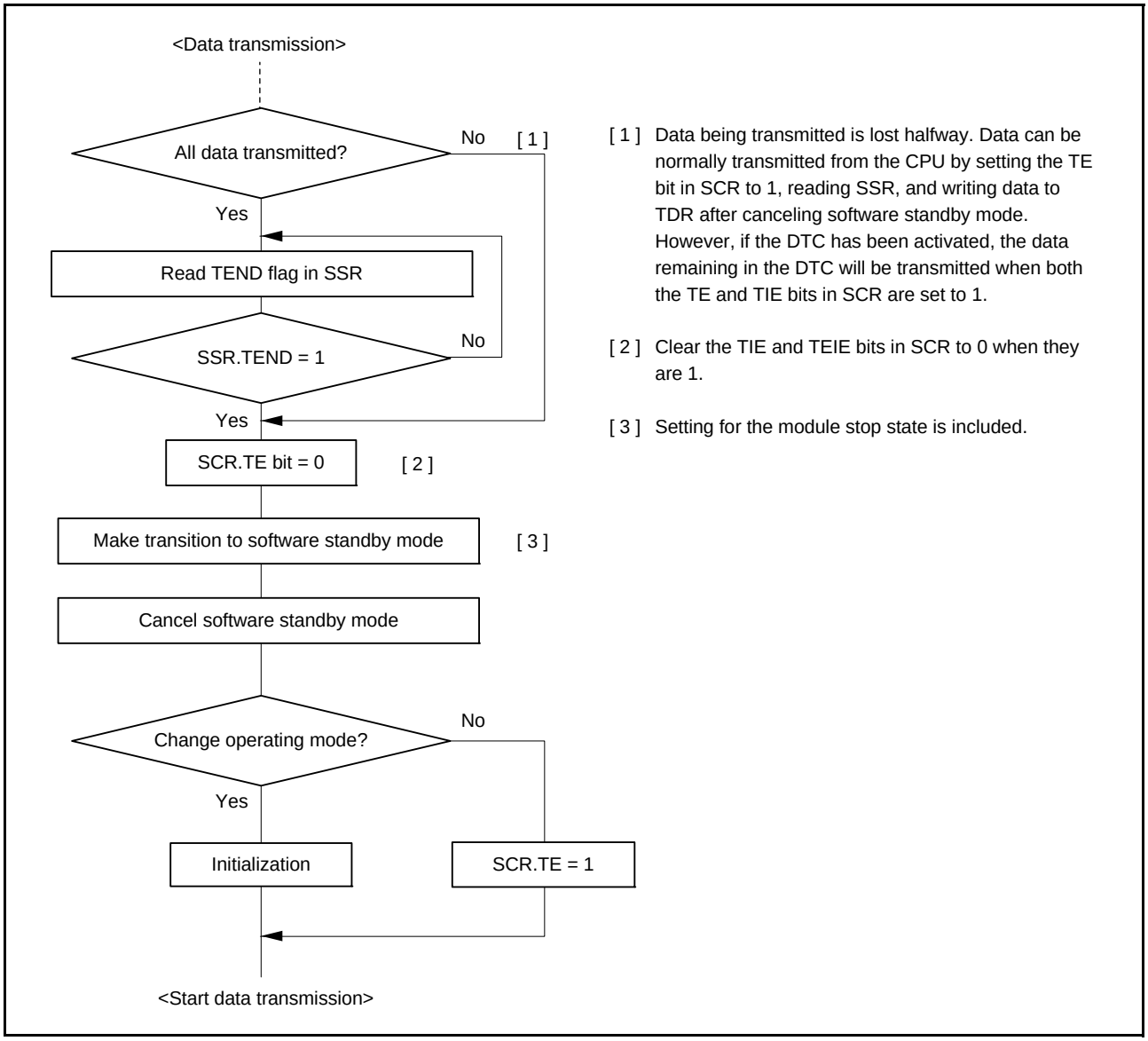


Figure 34.69 Example of Flowchart for Transition to Software Standby Mode during Transmission

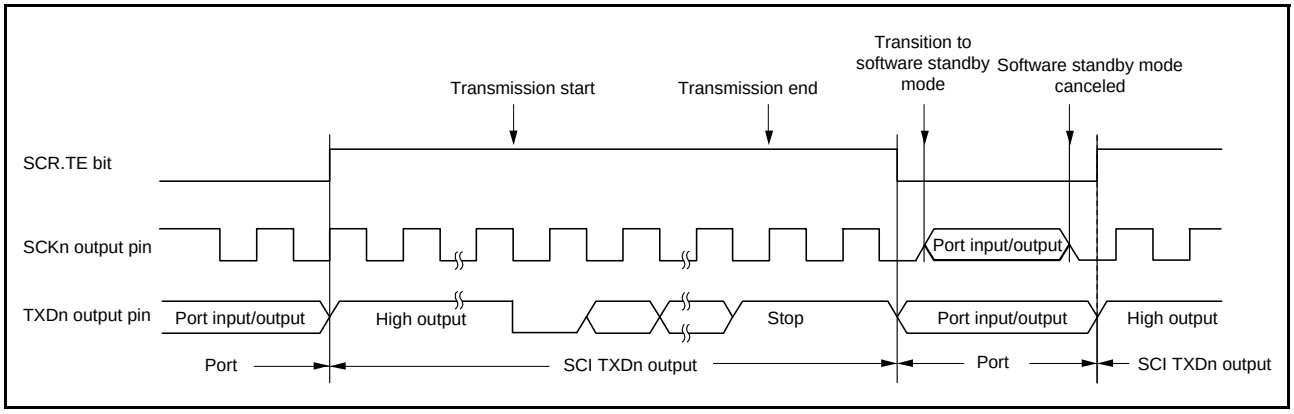


Figure 34.70 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

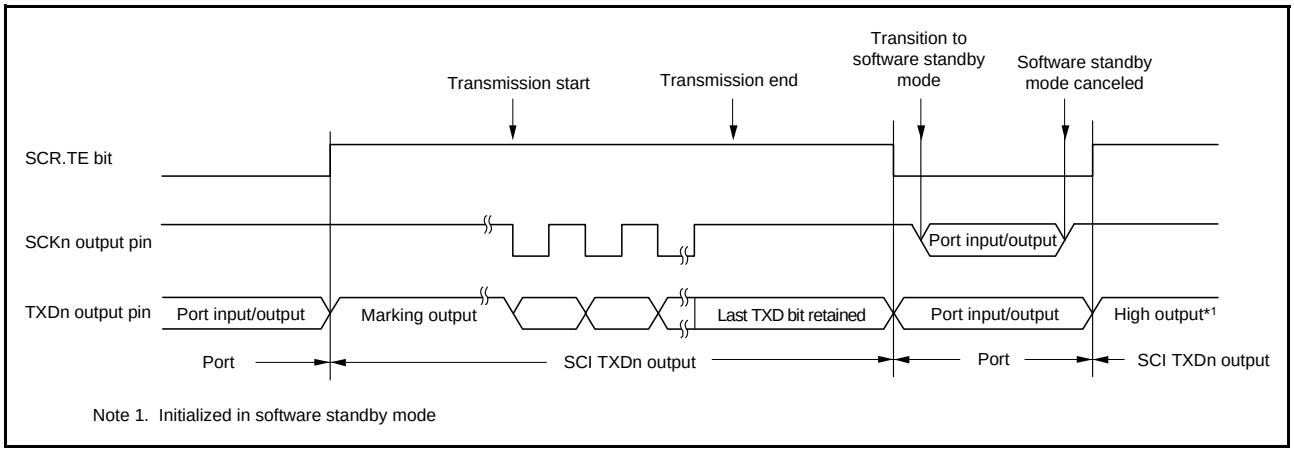


Figure 34.71 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

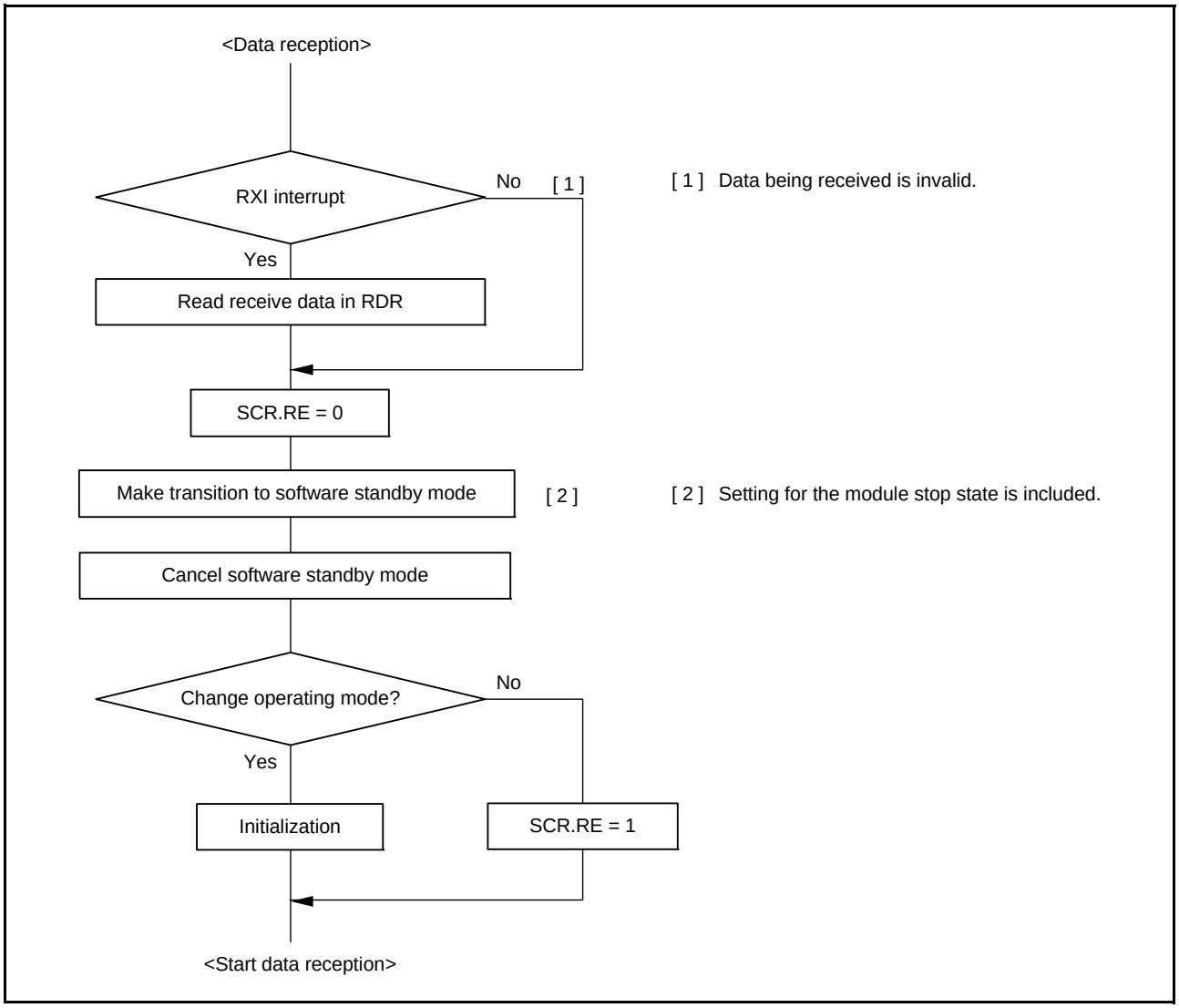


Figure 34.72 Example of Flowchart for Transition to Software Standby Mode during Reception

34.12.10 External Clock Input in Clock Synchronous Mode

In clock synchronous 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

34.12.11 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 CKPH and CKPOL bits in SPMR.
- In the case of the setting for clock delay (the SPMR.CKPH bit is 1), the received data full interrupt (RXI interrupt) is generated before the final clock edge on the SCKn pin as indicated in Figure 34.73. Take care because the output on the SCKn pin becomes high impedance when the TE and RE bits in the SCR are set to 0 at this time, and if this is done immediately, it can shorten the width of the clock pulse in the final cycle of the transfer clock. Also take care because the high level is applied to the SSn# pin input of a connected slave immediately after the RXI interrupt and this may lead to incorrect operation of the slave.
- When operation is in multi-master mode, take care because the SCKn pin output becomes high impedance while the input on the SS# 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

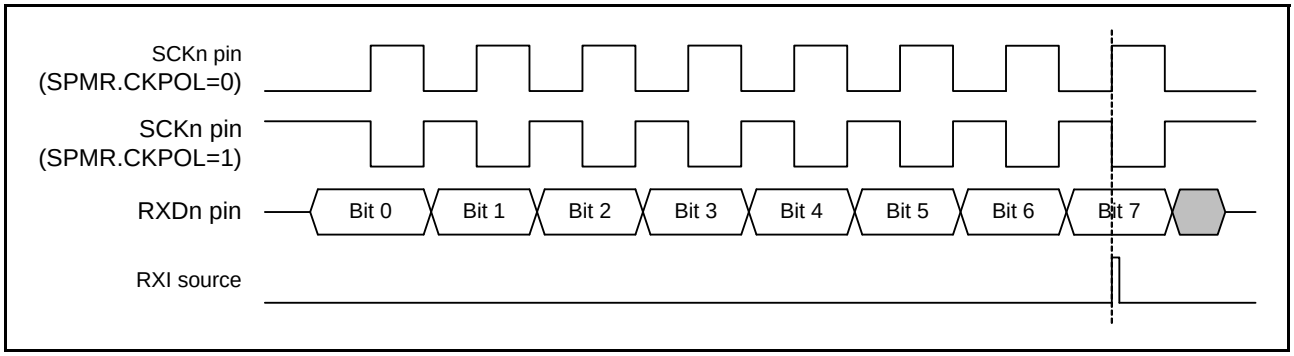


Figure 34.73 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- When data for transmission are written to the TDR, secure at least five cycles of the PCLK from input of the low level on the SSn# pin to input of the external clock.
- Provide an external clock signal to the master the same as the data length for transfer.
- 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.

34.12.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the SHARPS bit in PCR is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer of the SCId module is in Break Field low width output mode and the value of the TCST bit in TCR 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 TE bit in SCI12.SCR is 1.

34.12.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

An SCIc interrupt request is generated even if the extended serial mode is enabled. However, the SCIc interrupt should not be used during reception of a start frame because SCId uses an SCIc interrupt request.
The two ways of dealing with this are described below. When a reception error is detected, clear the error flag of the SCIc and initialize the control section of the SCId following the example of flowchart shown in Figure 34.74.

(1) Set the SCR.RIE bit of the SCIc to 0 to disable the output of interrupt requests. Check the error flags in the SSR register for SCIc on completion of the reception of a start frame, because an ERI interrupt is not generated if a reception error occurs. After reception of the start frame is completed, set the SCR.RIE bit of the SCIc to 1 by the time the first byte of the information frame is received.

(2) Set the SCR.RIE bit of the SCIc 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.

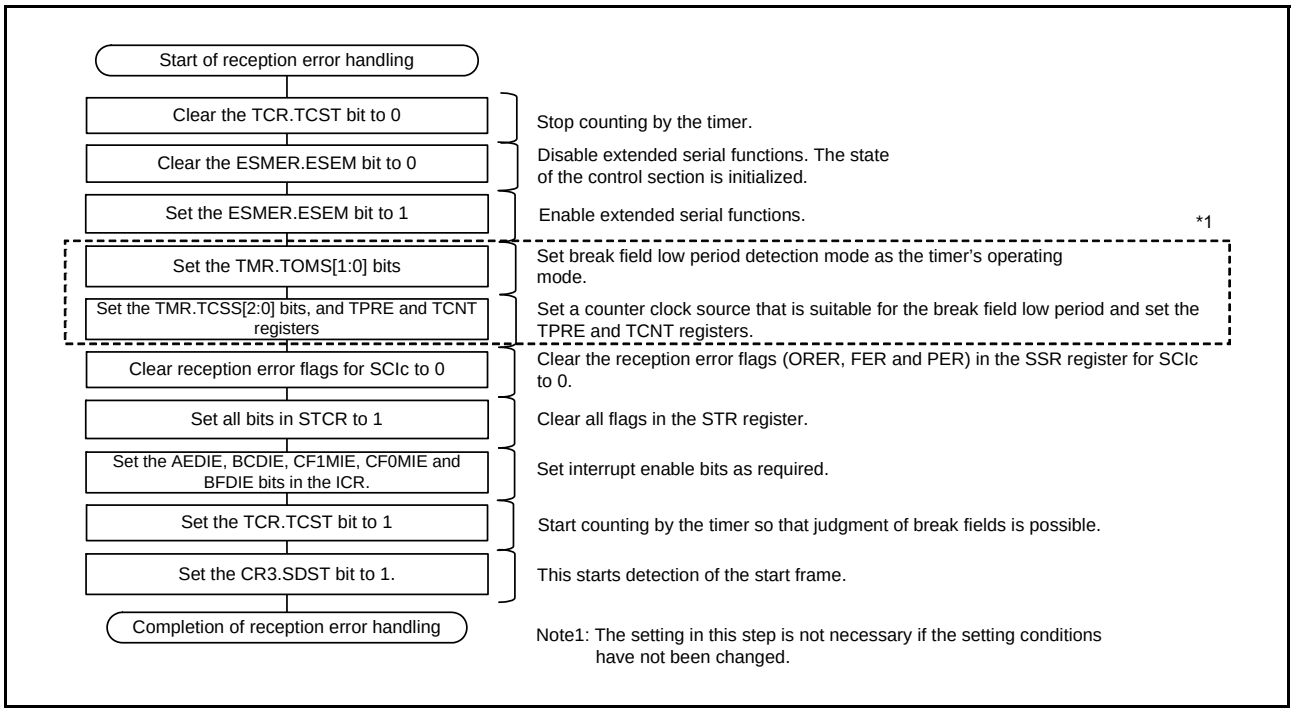


Figure 34.74 Example of Flowchart for Reception Error Handling (During Reception of the Start Frame)

35. I2C Bus Interface (RIIC)

The RX63N/RX631 Group has four I2C bus interfaces (RIIC modules).
The RIIC module conforms with and provides a subset of the NXP I2C bus (Inter-IC-Bus) interface functions.

35.1 Overview

Table 35.1 lists the specifications of the RIIC, Figure 35.1 shows a block diagram of the RIIC, and Figure 35.2 shows an example of I/O pin connections to external circuits (I2C bus configuration example). Table 35.2 lists the I/O pins of the RIIC.

Table 35.1 RIIC Specifications

Item	Specifications
Communications format	• I2C bus format or SMBus format • Master mode or slave mode selectable • Automatic securing of the various set-up times, hold times, and bus-free times for the transfer rate
Transfer rate	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 slave-address settings can be made. • Seven- and ten-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.
Acknowledgement	• 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 clock signal (SCL) 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 (WAIT function)
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) signal 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 time-out function is capable of detecting long-interval stop of the SCL (clock signal).
Noise removal	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 AL, NACK, time-out, 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) Transmission complete
Low power consumption function	Module stop state can be set.

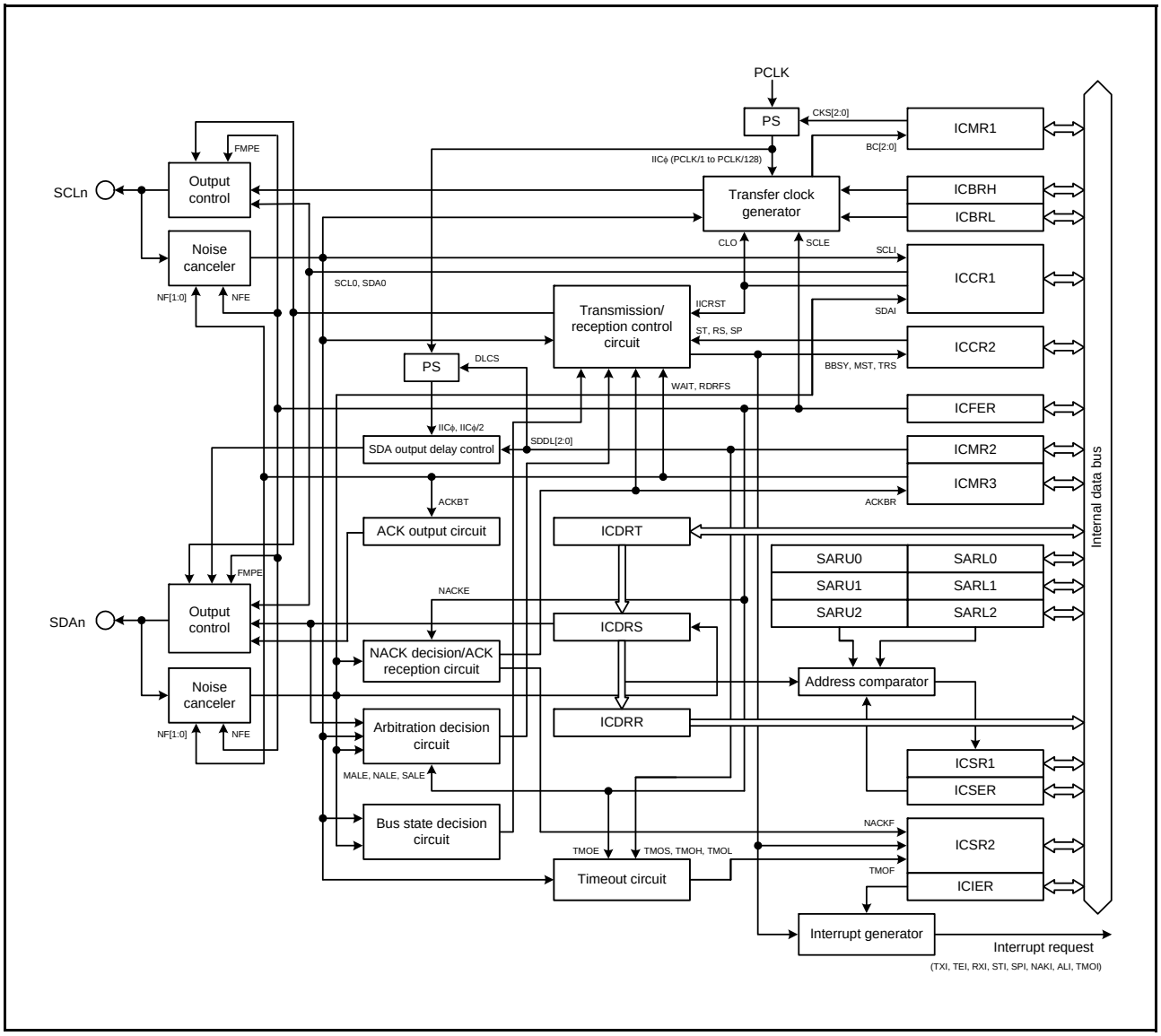


Figure 35.1 Block Diagram of RIIC

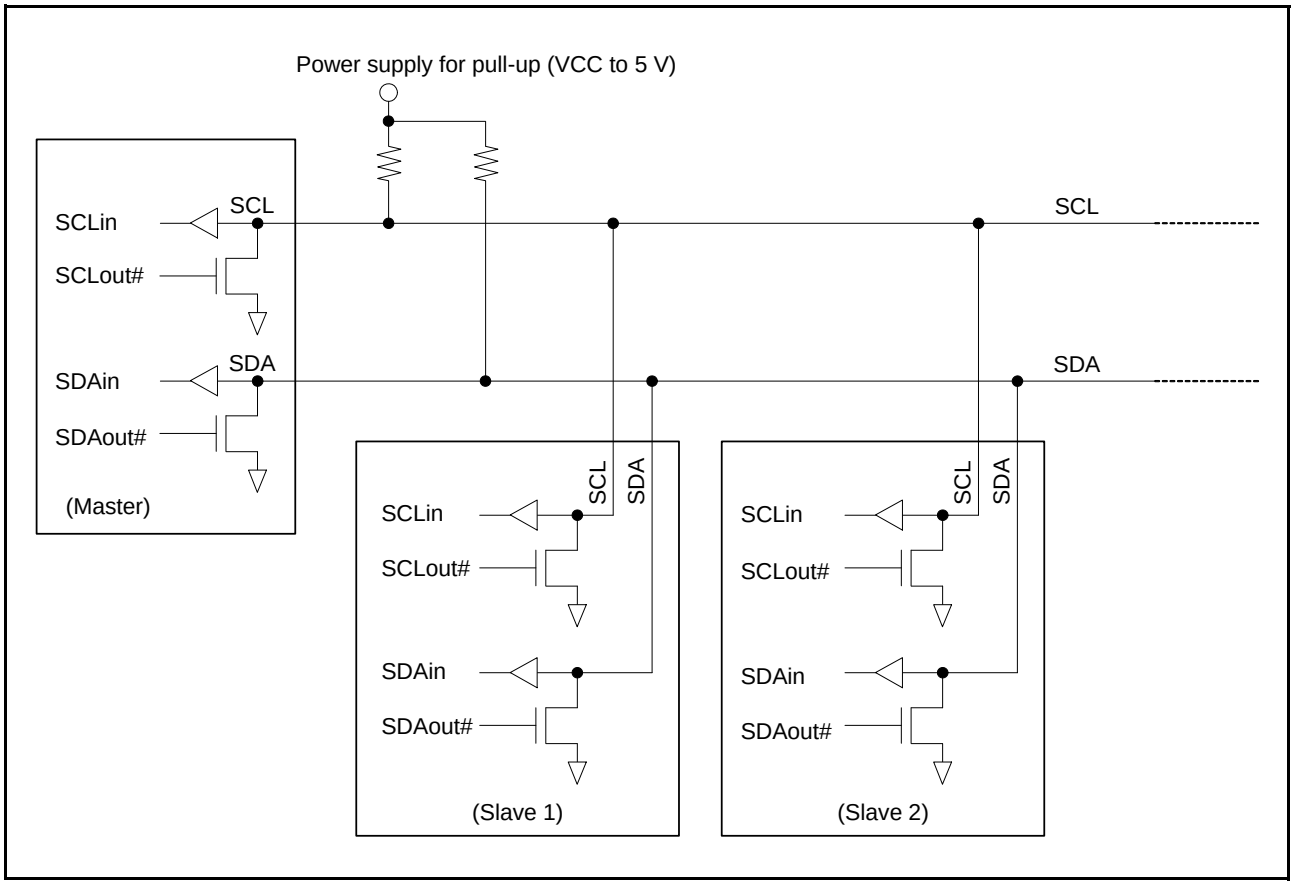


Figure 35.2 Connections to the External Circuit by the I/O Pins (I2C Bus Configuration Example)

The input level of the signals for RIIC is CMOS when I2C bus is selected (the ICMR3.SMBS bit = 0), or TTL when SMBus is selected (the ICMR3.SMBS bit = 1).

Table 35.2 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
RIIC1	SCL1	I/O	RIIC1 serial clock I/O pin
	SDA1	I/O	RIIC1 serial data I/O pin
RIIC2	SCL2	I/O	RIIC2 serial clock I/O pin
	SDA2	I/O	RIIC2 serial data I/O pin
RIIC3	SCL3	I/O	RIIC3 serial clock I/O pin
	SDA3	I/O	RIIC3 serial data I/O pin

35.2 Register Descriptions

35.2.1 I2C Bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h, RIIC1.ICCR1 0008 8320h, RIIC2.ICCR1 0008 8340h, RIIC3.ICCR1 0008 8360h

	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 Bus Input Monitor	0: SDAn pin input is at a low level. 1: SDAn pin input is at a high level.	R
b1	SCLI	SCL Bus Input Monitor	0: SCLn pin input is at a low level. 1: SCLn pin input is at a high level.	R
b2	SDAO	SDA Output Control	• Read: 0: SDAn pin output is at a low level. 1: SDAn pin is in a high-impedance state. • Write: 0: Changes the SDAn pin output to a low level. 1: Changes the SDAn pin in a high-impedance state. (High level output is achieved through an external pull-up resistor.)	R/W *1,*2
b3	SCLO	SCL Output Control	• Read: 0: SCLn pin output is at a low level. 1: SCLn pin is in a high-impedance state. • Write: 0: Changes the SCLn pin output to a low level. 1: Changes the SCLn pin in a high-impedance state. (High level output is achieved through an external pull-up resistor.)	R/W *1,*2
b4	SOWP	SCLO/SDAO Write Protect	0: Allows the SCLO and SDAO bits to be rewritten. (This bit is read as 1.)	R/W*2
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: Clears 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: Disables the RIIC (the SCLn pin SDAn pins are inactive). 1: Enables the RIIC (the SCLn and SDAn pins are active). (Selects RIIC reset or internal reset in the combination with the IICRST bit)	R/W

Note 1. Do not write to these bits during communication. Changing a value during communication may cause a transmission or reception failure or an AL error.
Note 2. To change the SDAO and SCLO bits, set the SOWP bit to 0 at the same timing to set the SDAO and SCLO bits to 0.

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, see section 35.11.2, 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 35.3 lists the resets of the RIIC.
The RIIC reset resets all registers including the BBSY flag in ICCR2 and internal states of the RIIC, and the internal reset resets the bit counter (BC[2:0] bits in ICMR1), 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, see section 35.14, Reset States.
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 35.3 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Reset the BC[2:0] bits in ICMR1, and the ICSR1, ICSR2, ICDRS registers and the internal states of the RIIC.

ICE Bit (I2C Bus Interface Enable)

This pin selects the active or inactive state of the SCLn and SDAn pins. The ICE bit is also used to provide two types of resets in combination with the IICRST bit. For the types of resets, see Table 35.3, RIIC Resets.
Set the ICE bit to 1 when using the RIIC. When the ICE bit is set to 1, the SCLn and SDAn pins are placed in the active state.
Set the ICE bit to 0 when not using the RIIC. When the ICE bit is set to 0, the SCLn and SDAn pins are placed in the inactive state.
Do not assign the SCLn and SDAn pins to the RIIC when configuring the multifunction pin controller (MPC). Note that slave address comparisons will be performed if these pins are assigned to the RIIC.

35.2.2 I2C Bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h, RIIC1.ICCR2 0008 8321h, RIIC2.ICCR2 0008 8341h, RIIC3.ICCR2 0008 8361h

	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: Transmit 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 or in the bus free state).	R

Note 1. When the MTWP bit in ICMR1 is set to 1, the MST and TRS bits 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).

For details on the start condition issuance, see section 35.10, 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
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 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).
Note that arbitration may be lost if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

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) and the MST bit is set to 1 (master mode).
For details on the restart condition issuance, see section 35.10, Start Condition/Restart Condition/Stop Condition Issuing Function.
[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 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 the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the restart condition issuance request bit remains set. If the operating mode changes to master mode with the bit not being cleared, 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) and the MST bit is set to 1 (master mode).
For details on the stop condition issuance, see section 35.10, 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 MST bit in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the SP bit after reading SP = 1
- When a stop condition has been issued or a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 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).
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 the TRS bit is automatically changed to the value for transmission mode or reception mode by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 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 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 ICSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- The AL (arbitration-lost) flag in ICSR2 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 ICSER 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 = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 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 the value for master mode or slave mode by detection or issuing of a start condition, etc. Although writing to the MST bit is possible when the MTWP bit in ICMR1 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 MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I2C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDAnn line changes from high to low under the condition of SCLn = high, assuming that a start condition has been issued.

When the SDAn line changes from low to high under the condition of SCLn = high, this bit is cleared to 0 after the bus free time (specified in ICBRL) 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 ICBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (RIIC reset)

35.2.3 I2C Bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h, RIIC1.ICMR1 0008 8322h, RIIC2.ICMR1 0008 8342h, RIIC3.ICMR1 0008 8362h

	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 Selection	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 MST and TRS bits in ICCR2. 1: Enables writing to the MST and TRS bits in ICCR2.	R/W

Note 1. Set the BCWP bit to 0 to rewrite the BC[2:0] bits. The BC[2:0] bits must be rewritten by using the MOV instruction.

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 frames 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.

35.2.4 I²C Bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h, RIIC1.ICMR2 0008 8323h, RIIC2.ICMR2 0008 8343h, RIIC3.ICMR2 0008 8363h

	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 Selection	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count is disabled while the SCLn line is at a low level. 1: Count is enabled while the SCLn line is at a low level.	R/W
b2	TMOH	Timeout H Count Control	0: Count is disabled while the SCLn line is at a high level. 1: Count 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 = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycle 0 1 1: 3 IICϕ cycle 1 0 0: 4 IICϕ cycle 1 0 1: 5 IICϕ cycle 1 1 0: 6 IICϕ cycle 1 1 1: 7 IICϕ cycle • When ICMR2.DLCS = 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 Selection	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.	R/W

TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (TMOE bit = 1 in ICFER). 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, see section 35.11.1, 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 (TMOE bit = 1 in ICFER).

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 (TMOE bit = 1 in ICFER).

SDDL[2:0] Bits (SDA Output Delay Setup 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.

For details on this function, see section 35.5, Facility for Delaying SDA Output.

Note: • Set the SDA output delay time to meet the I2C bus standard (within the data enable time/acknowledge enable time*1) or the SMBus standard (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 standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

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+])

35.2.5 I2C Bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h, RIIC1.ICMR3 0008 8324h, RIIC2.ICMR3 0008 8344h, RIIC3.ICMR3 0008 8364h

	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 Selection	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: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 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.	W*1
b5	RDRFS	RDRF Flag Set Timing Selection	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 ICDRR.	R/W*2
b7	SMBS	SMBus/I2C Bus Selection	0: The I2C bus is selected. 1: The SMBus is selected.	R/W

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot 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 Selection)

These bits are used to select the number of stages of the digital noise filter.

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 range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock (IICϕ) cycles + analog noise filter: 120 ns (reference values)] or more, 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 TRS bit in ICCR2 set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 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 SP bit in ICCR2 set to 1)
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (RIIC reset)

Note: • The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Selection)

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 = 0) or NACK (ACKBT = 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 receive data buffer (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 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 beforehand.

SMBS Bit (SMBus/I2C Bus Selection)

Setting this bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

35.2.6 I2C Bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h, RIIC1.ICFER 0008 8325h, RIIC2.ICFER 0008 8345h, RIIC3.ICFER 0008 8365h

	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 MST and TRS bits in ICCR2 automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the MST and TRS bits in ICCR2 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 RIIC1 to RIIC3, 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, see section 35.11.1, 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, see section 35.8.2, 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 cleared to 0 (SCL synchronous circuit not 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 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 cleared to 0 except for checking the output of the transfer rate that was set during debugging.

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 standard (tof) of the I2C bus is selected. When this bit is cleared to 0, a slope control circuit conforming to the Standard-mode[Sm] and Fast-mode[fm] slope control standard (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 standard. Clear 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).

35.2.7 I2C Bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h, RIIC1.ICSER 0008 8326h, RIIC2.ICSER 0008 8346h, RIIC3.ICSER 0008 8366h

	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 SARL0 and SARU0 is disabled. 1: Slave address in SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in SARL1 and SARU1 is disabled. 1: Slave address in SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in SARL2 and SARU2 is disabled. 1: Slave address in 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 received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in 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 [W]: 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 SARLy and SARUy (y = 0 to 2) and performs data receive operation.

When this bit is cleared 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 frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see section 35.7.3, 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 SMBS bit in ICMR3 is 1. When this bit is set to 1 while the SMBS bit in ICMR3 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 SARLy and SARUy (y = 0 to 2) and performs the receive operation.
When the SMBS bit in ICMR3 or the HOAE bit is cleared to 0, the received slave address is ignored even if it matches the host address.

35.2.8 I2C Bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h, RIIC1.ICIER 0008 8327h, RIIC2.ICIER 0008 8347h, RIIC3.ICIER 0008 8367h

	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 Enable	0: Timeout interrupt request (TMOI) is disabled. 1: Timeout interrupt request (TMOI) is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Enable	0: Arbitration-lost interrupt request (ALI) is disabled. 1: Arbitration-lost interrupt request (ALI) is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Enable	0: Start condition detection interrupt request (STI) is disabled. 1: Start condition detection interrupt request (STI) is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Enable	0: Stop condition detection interrupt request (SPI) is disabled. 1: Stop condition detection interrupt request (SPI) is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Enable	0: NACK reception interrupt request (NAKI) is disabled. 1: NACK reception interrupt request (NAKI) is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt request (ICRXI) is disabled. 1: Receive data full interrupt request (ICRXI) is enabled.	R/W
b6	TEIE	Transmit End Interrupt Enable	0: Transmit end interrupt request (ICTEI) is disabled. 1: Transmit end interrupt request (ICTEI) is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt request (ICTXI) is disabled. 1: Transmit data empty interrupt request (ICTXI) is enabled.	R/W

TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the TMOF flag in ICSR2 is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Enable)

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the AL flag in ICSR2 is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt requests (STI) when the START flag in ICSR2 is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the STOP flag in ICSR2 is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the NACKF flag in ICSR2 is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Enable)

This bit is used to enable or disable receive data full interrupt requests (ICRXI) when the RDRF flag in ICSR2 is set to 1.

TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (ICTEI) when the TEND flag in ICSR2 is set to 1. An ICTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (ICTXI) when the TDRE flag in ICSR2 is set to 1.

35.2.9 I2C Bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h, RIIC1.ICSR1 0008 8328h, RIIC2.ICSR1 0008 8348h, RIIC3.ICSR1 0008 8368h

	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. • This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL0 while the FS bit in SARU0 is 0 (7-bit address format selected). • This bit is set to 1 when the received slave address matches a value of (11110b + SVA[1:0] in SARU0) and the following address matches the SARL0 value while the FS bit in SARU0 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL0 match determination frame.)	R(/W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected. • This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL1 while the FS bit in SARU1 is 0 (7-bit address format selected). • This bit is set to 1 when the received slave address matches a value of (11110b + SVA[1:0] in SARU1) and the following address matches the SARL1 value while the FS bit in SARU1 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL1 match determination frame.)	R(/W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected. • This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL2 while the FS bit in SARU2 is 0 (7-bit address format selected). • This bit is set to 1 when the received slave address matches a value of (11110b + SVA[1:0] in SARU2) and the following address matches the SARL2 value while the FS bit in SARU2 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL2 match determination frame.)	R(/W) *1

Bit	Symbol	Bit Name	Description	R/W
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected. • This bit is set to 1 when the received slave address matches the general call address (all 0).	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 frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]).	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) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: SARUy.FS = 0>

- When the received slave address matches the SVA[6:0] value in SARLy with the SARyE bit in ICSER 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 frame.

<For 10-bit address format: SARUy.FS = 1>

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy) and the following address matches the SARLy value with the SARyE bit in ICSER 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 frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

<For 7-bit address format: SARUy.FS = 0>

- When the received slave address does not match the SVA[6:0] value in SARLy with the SARyE bit in ICSER set to 1 (slave address m detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: SARUy.FS = 1>

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy) with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy) and the following address does not match the SARLy value with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSER set to 1 (general call address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSER set to 1 (general call address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) with the DIDE bit in ICSER set to 1 (Device-ID address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the DIDE bit in ICSER set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the DIDE bit in ICSER set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection)

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the HOAE bit in ICSER set to 1 (host address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA bit after reading HOA = 1
- When a stop condition is detected
- When 0 is written to the SMBS bit in ICMR3 or the HOAE bit in ICSER
- When the received slave address does not match the host address (0001 000b) with the HOAE bit in ICSER set to 1 (host address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

35.2.10 I2C Bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h, RIIC1.ICSR2 0008 8329h, RIIC2.ICSR2 0008 8349h, RIIC3.ICSR2 0008 8369h

	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: ICDRR contains no receive data. 1: ICDRR 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: ICDRT contains transmit data. 1: ICDRT contains no transmit data.	R

Note 1. Only 0 can be written to clear the flag.

TMOF Flag (Timeout Detection)

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 TMOH, TMOL, and TMOS in ICMR2 with the TMOE bit in ICFER set to 1 (timeout detection enabled) in master mode or in the slave specification state.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost)

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 bit 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 ST bit in ICCR2 is 1 (start condition issuance request) or the internal SDA output state does not match the SDAn line level
- When the ST bit in ICCR2 is set to 1 (start condition issuance request) with the BBSY flag in ICCR2 set to 1.

<When NACK 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 in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled>

- 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 bit after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Table 35.4 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 ST bit in ICCR2 is 1
					When ST in ICCR2 is set to 1 with BBSY in ICCR2 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)

[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 IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection)

[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 IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection)

[Setting condition]

- When acknowledge is not received (NACK is received) from the receive device in transmit mode with the NACKE bit in ICFER 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 IICRST bit in ICCR1 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 ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

RDRF Flag (Receive Data Full)

[Setting conditions]

- When receive data has been transferred from ICDRS to ICDRR
  This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the RDRFS bit in ICMR3)
- When the received slave address matches after a start condition (or a restart condition) is detected with the TRS bit in ICCR2 cleared to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

TEND Flag (Transmit End)

[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 ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty)

[Setting conditions]

- When data has been transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
  a. When the MST bit in ICCR2 is set to 1 after a start condition (or a restart condition) is detected
  b. When the RIIC enters transmit mode from receive mode
  c. When 1 is written to while the ICMR1.MTWP bit is 1
- When the received slave address matches while the TRS bit is 1

[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is cleared to 0
  a. When a stop condition is detected
  b. When the RIIC enters receive mode from transmit mode
  c. When 0 is written to while the ICMR1.MTWP bit is 1

- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • When the NACKF flag is set to 1 while the NACKE bit in ICFER 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.

35.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC1.SARL0 0008 832Ah, RIIC2.SARL0 0008 834Ah, RIIC3.SARL0 0008 836Ah
RIIC0.SARL1 0008 830Ch, RIIC1.SARL1 0008 832Ch, RIIC2.SARL1 0008 834Ch, RIIC3.SARL1 0008 836Ch
RIIC0.SARL2 0008 830Eh, RIIC1.SARL2 0008 832Eh, RIIC2.SARL2 0008 834Eh, RIIC3.SARL2 0008 836Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	The least significant bit (LSB) of a 10-bit slave address is set. • When the FS bit in SARUy is 0 (7-bit address format), this bit is invalid. • When the FS bit in SARUy is 1 (10-bit address format), this bit is the LSB of the lower 8-bit address (combined with the SVA[6:0] bits) of a 10-bit slave address.	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	A slave address is set. • When the FS bit in SARUy is 0 (7-bit address format), these bits form a 7-bit slave address. • When the FS bit in SARUy is 1 (10-bit address format), these bits form the lower 8-bit address (combined with the SVA0 bit) of a 10-bit slave address.	R/W

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), this bit functions as the LSB of a 10-bit address and forms the lower eight bits of a 10-bit address in combination with the SVA[6:0] bits.
When the SARyE bit in ICSER 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 = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits function as the lower eight bits of a 10-bit address in combination with the SVA0 bit.
While the SARyE bit in ICSER is 0, the setting of these bits is ignored.

35.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC1.SARU0 0008 832Bh, RIIC2.SARU0 0008 834Bh, RIIC3.SARU0 0008 834Bh
RIIC0.SARU1 0008 830Dh, RIIC1.SARU1 0008 832Dh, RIIC2.SARU1 0008 834Dh, RIIC3.SARU1 0008 836Dh
RIIC0.SARU2 0008 830Fh, RIIC1.SARU2 0008 832Fh, RIIC2.SARU2 0008 834Fh, RIIC3.SARU2 0008 836Fh

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

Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Selection	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. • When the SARUy.FS bit is 0 (7-bit address format), these bits are invalid. • When the SARUy.FS bit is 1 (10-bit address format), these bits form the upper two bits of a 10-bit slave address.	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 Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in SARLy and SARUy).
When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the settings of the SVA[1:0] bits and the SVA0 bit in SARLy are ignored.
When the SARyE bit in ICSER is set to 1 (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 SARyE bit in ICSER is 0 (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 two bits of a 10-bit address.
When the SARyE bit in ICSER 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.

35.2.13 I2C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h, RIIC1.ICBRL 0008 8330h, RIIC2.ICBRL 0008 8350h, RIIC3.ICBRL 0008 8370h

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

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

It also works to generate the data setup time for automatic SCL low-hold operation (see section 35.8, Automatically 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 source (IICϕ) specified by the CKS[2:0] bits in ICMR1.

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+])

35.2.14 I2C Bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h, RIIC1.ICBRH 0008 8331h, RIIC2.ICBRH 0008 8351h, RIIC3.ICBRH 0008 8371h

	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 source (IICϕ) specified by the CKS[2:0] bits in ICMR1.

The I2C transfer rate and the SCL clock duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICϕ* + SCLn line rising time [tr] + SCLnn line falling time [tf]}
Duty cycle = {SCLn line rising time [tr]* + (ICBRH + 1) / IICϕ} / {SCLn line falling time [tf]* + (ICBRL + 1) / IICϕ}

IICϕ = PCLK × 10^6 × Division ratio
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 standard from NXP Semiconductors.

Table 35.5 lists examples of ICBRH/ICBRL settings.

Table 35.5 Examples of ICBRH/ICBRL Settings for Transfer Rate (1)

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)

Table 35.5 Examples of ICBRH/ICBRL Settings for Transfer Rate (2)

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)

Table 35.5 Examples of ICBRH/ICBRL Settings for Transfer Rate (3)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	30			33			50		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	26 (FAh)	111b	16 (F0h)	20 (F4h)
50	100b	15 (EFh)	18 (F2h)	100b	17 (F1h)	20 (F4h)	100b	26 (FAh)	31 (FFh)
100	010b	2 (E2h)	3 (E3h)	011b	16 (F0h)	19 (F3h)	011b	24 (F8h)	29 (FDh)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	21 (F5h)	010b	7 (E7h)	16 (F0h)
1000	000b	7 (E7h)	14 (EEh)	000b	8 (E8h)	16 (F0h)	000b	12 (ECh)	24 (F8h)

Note: • CBRH/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 standard from NXP Semiconductors.

35.2.15 I2C Bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h, RIIC1.ICDRT 0008 8332h, RIIC2.ICDRT 0008 8352h, RIIC3.ICDRT 0008 8372h

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

When ICDRT detects a space in the I2C bus shift register (ICDRS), it transfers the transmit data that has been written to ICDRT to ICDRS and starts transmitting data in transmit mode.
The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data has been written to ICDRT while the ICDRS data is being transmitted.
ICDRT can always be read and written. Write transmit data to ICDRT once when a transmit data empty interrupt (ICTXI) request is generated.

35.2.16 I2C Bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h, RIIC1.ICDRR 0008 8333h, RIIC2.ICDRR 0008 8353h, RIIC3.ICDRR 0008 8373h

	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 ICDRR to enable the next data to be received.
The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data has been read from ICDRR while ICDRS is receiving data.
ICDRR cannot be written. Read data from ICDRR once when a receive data full interrupt (ICRXI) request is generated.
If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the RIIC automatically holds the SCLn clock low one cycle before the RDRF flag is set to 1 next.

35.2.17 I2C Bus Shift Register (ICDRS)

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

ICDRS is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from ICDRT to ICDRS and is sent from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data has been received.
ICDRS cannot be accessed directly.

35.3 Operation

35.3.1 Communication Data Format

The I2C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame 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 35.3 shows the I2C bus format, and Figure 35.4 shows the I2C bus timing.

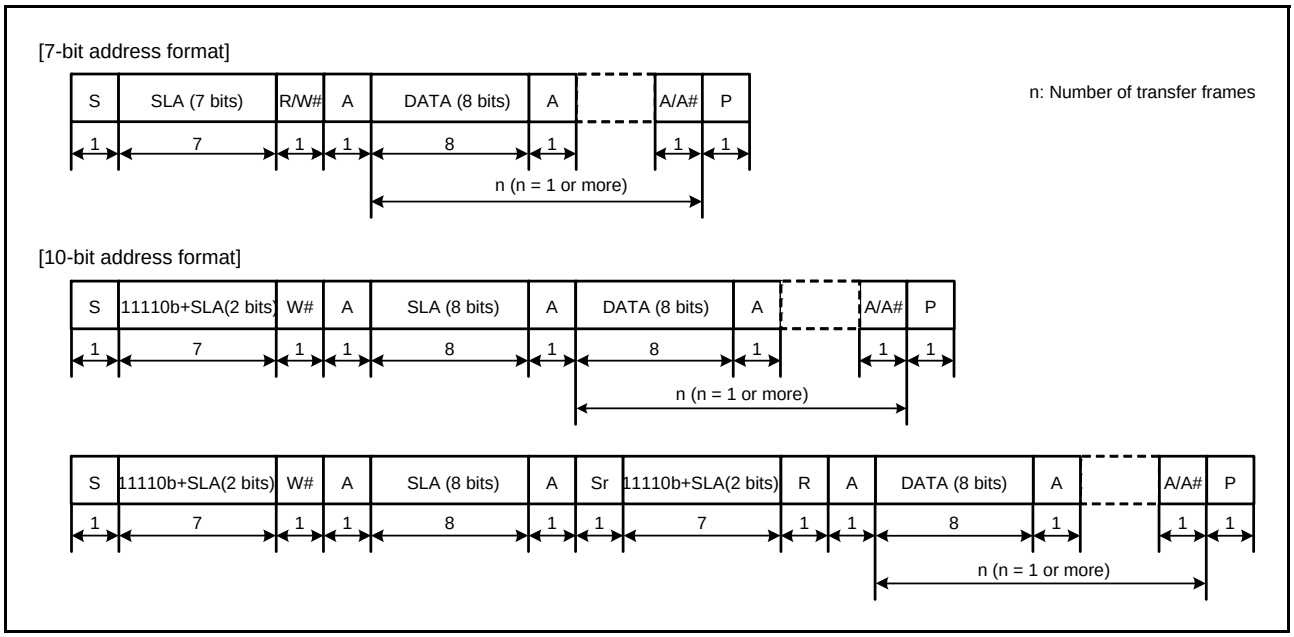


Figure 35.3 I2C Bus Format

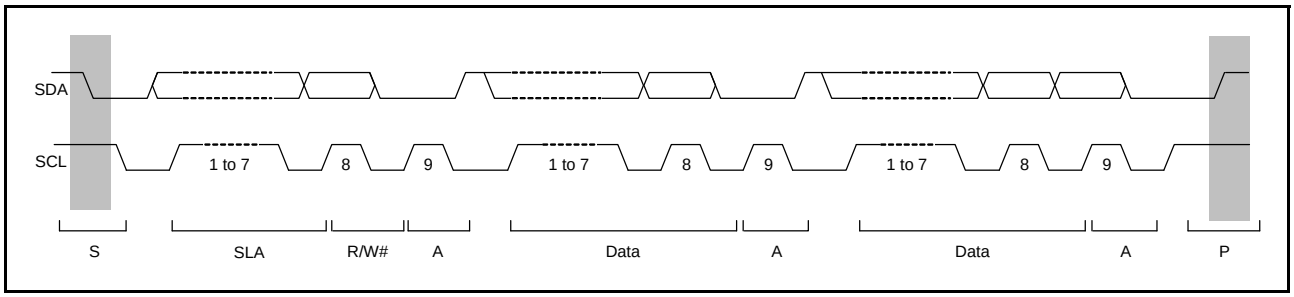


Figure 35.4 I2C Bus Timing (SLA = 7 Bits)

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.)

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.

35.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 35.5.

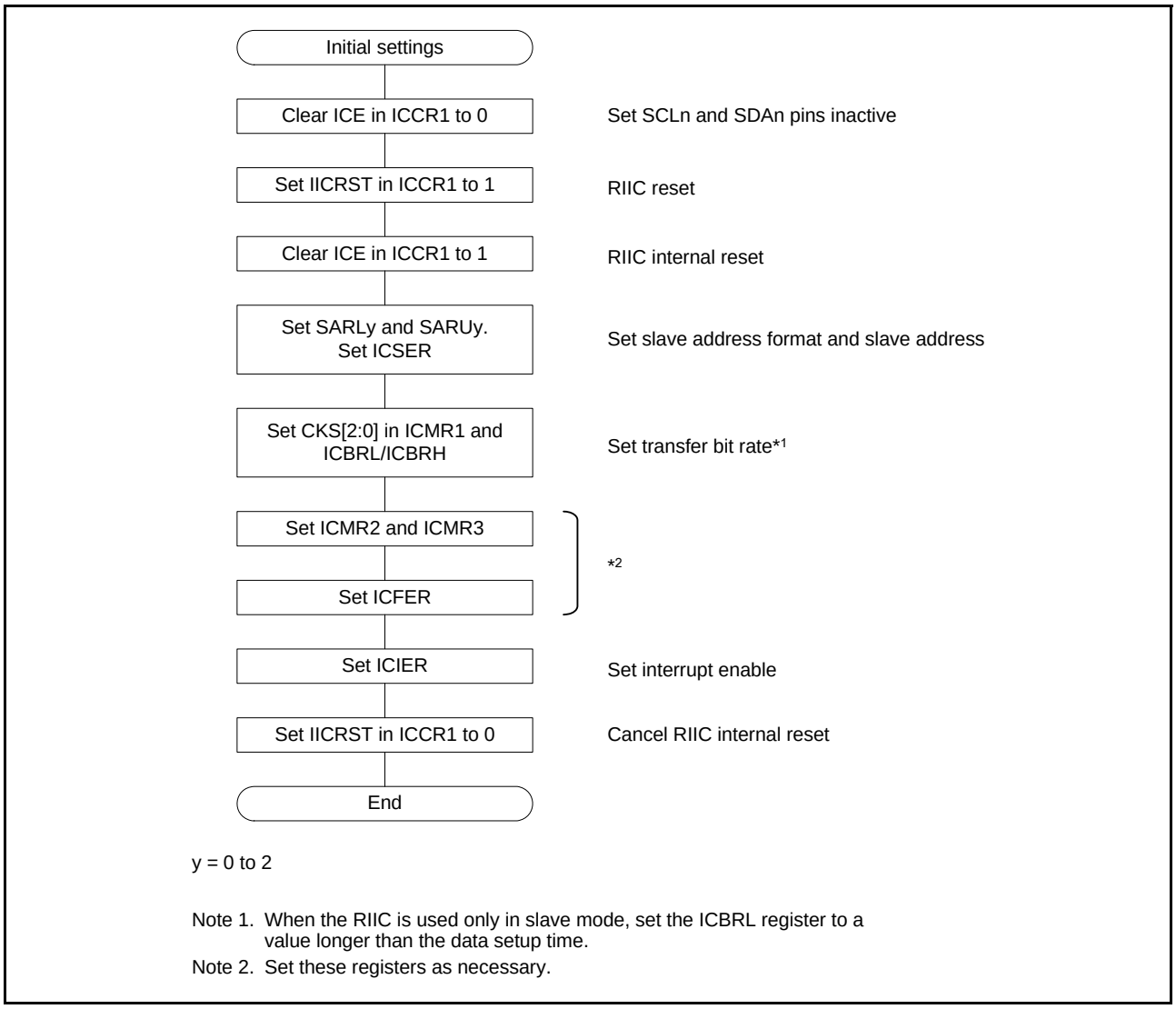


Figure 35.5 Example of RIIC Initialization Flow

35.3.3 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 acknowledgements. Figure 35.6 shows an example of usage of master transmission and Figure 35.7 to Figure 35.9 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) while keeping the ICCR1.ICE bit at 0 (SCLn and SDAn pins are in inactive state). This initializes the flag bits and internal state of the ICSR1 register. After that, set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2). Then, set the other resisters as necessary (see figure 35.5, Example of RIIC Initialization Flow, for the RIIC initialization procedure). When the necessary register settings have been completed, clear the ICCR1.IICRST bit to 0 (reset cancelation). This step is not necessary if the RIIC initialization has already been completed.

(2) Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 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 START flag in ICSR2 are automatically set to 1 and the ST bit is automatically cleared 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 the MST and TRS bits in ICCR2 are automatically set to 1, placing the RIIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, 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 transmitter 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.
Since 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 two higher-order bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to ICDRT.

(4) After confirming that the TDRE flag in ICSR2 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 TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 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 clears the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Furthermore, it automatically clears the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.

(7) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

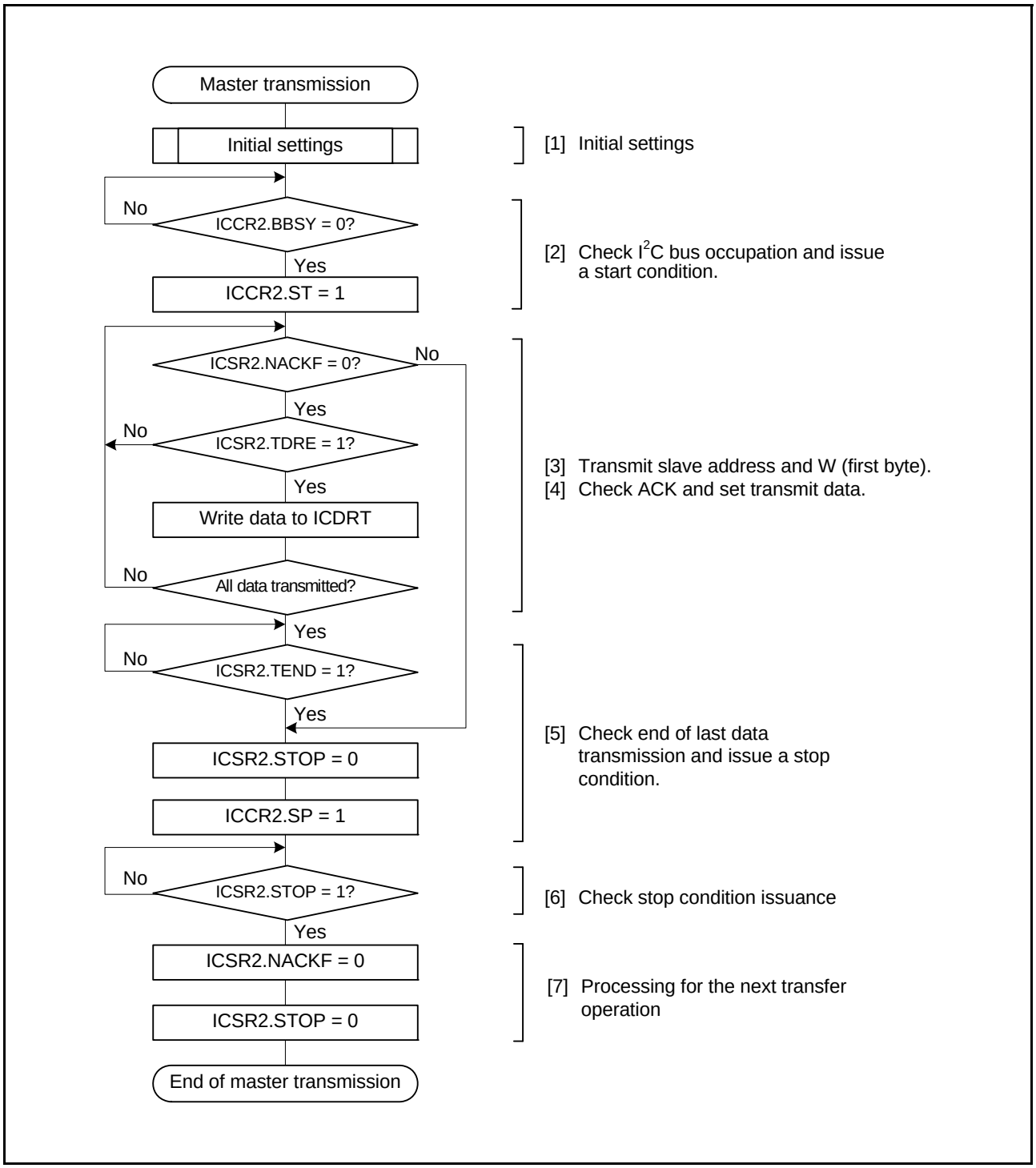


Figure 35.6 Example of Master Transmission Flowchart

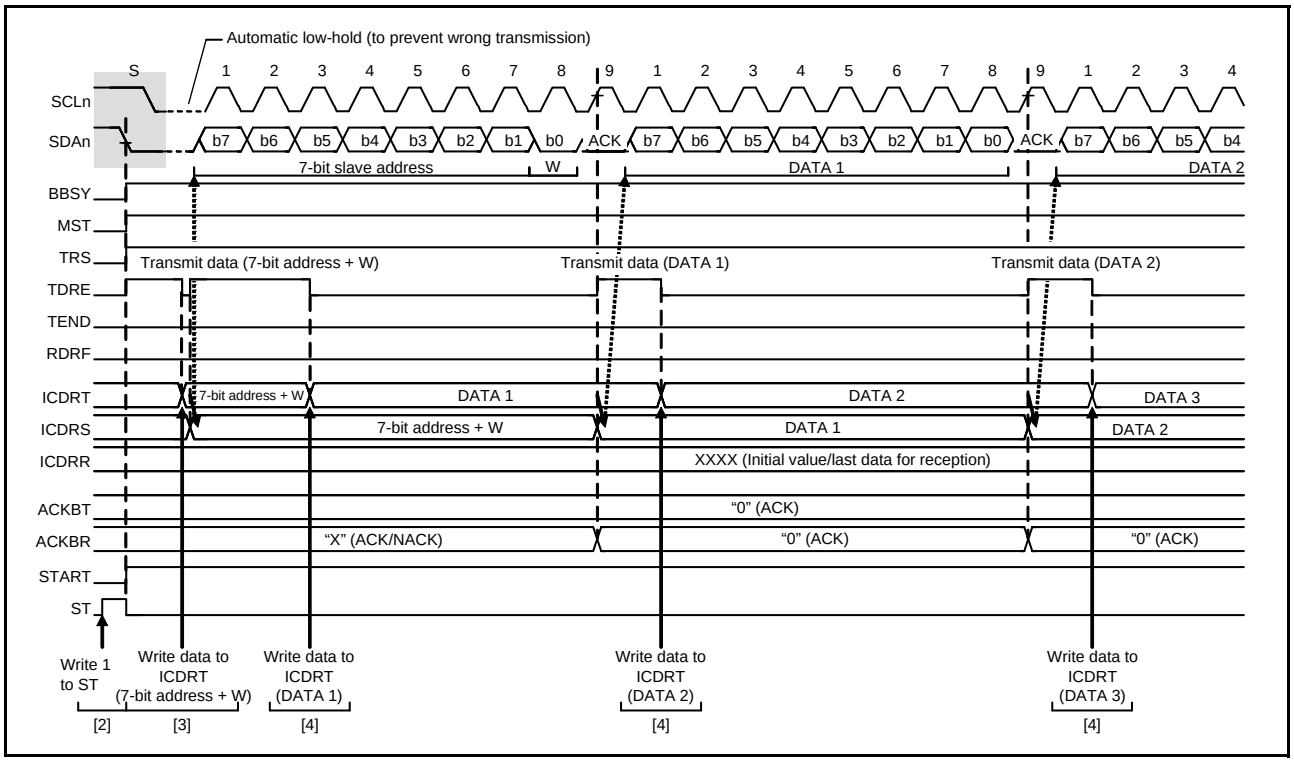


Figure 35.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

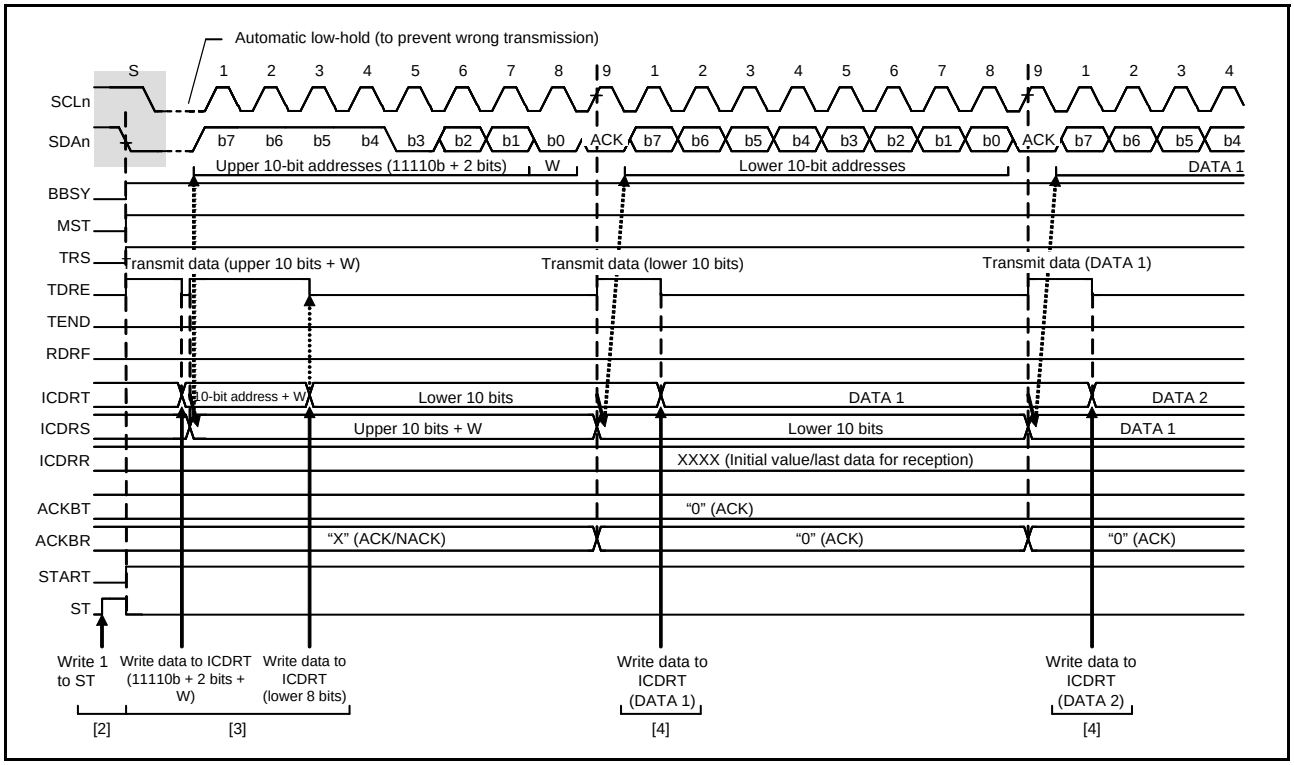


Figure 35.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

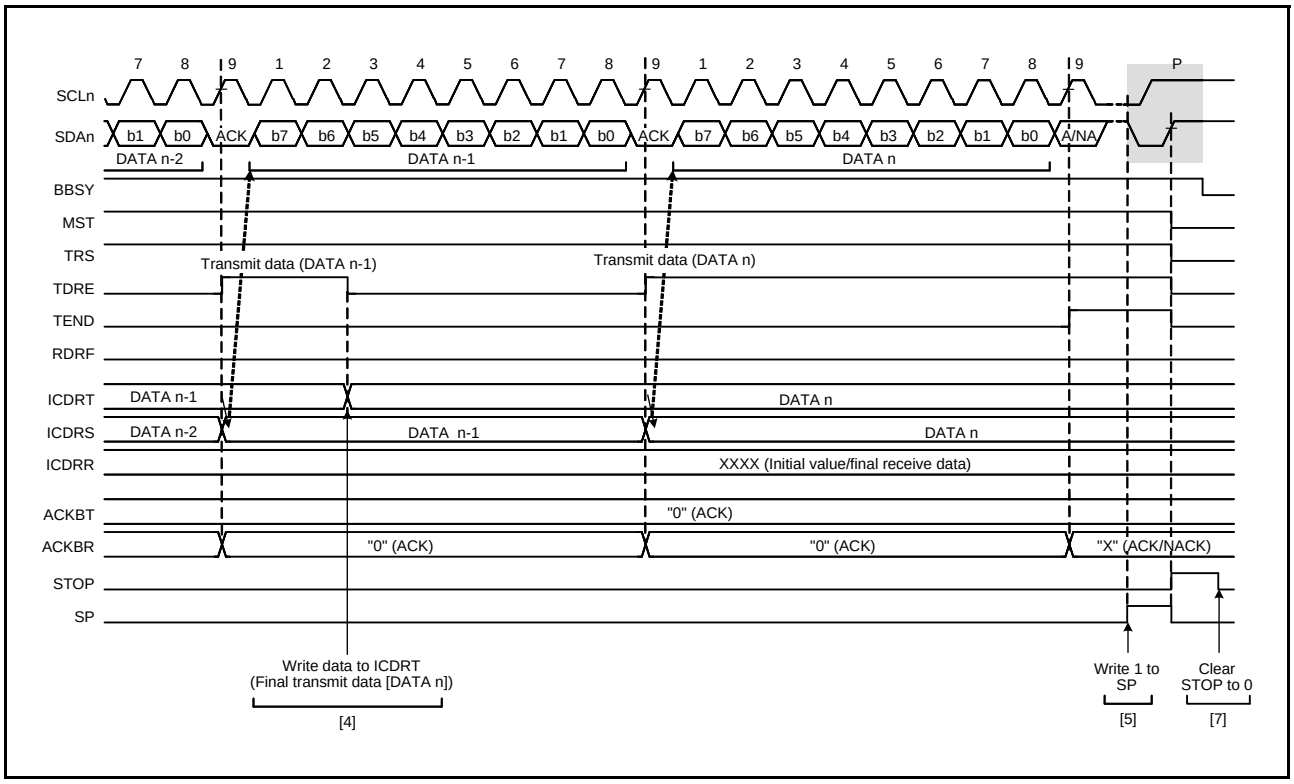


Figure 35.9 Master Transmit Operation Timing (3)

35.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since 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 35.10 shows an example of usage of master reception and Figure 35.11 and Figure 35.13 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

(1) Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) while keeping the ICCR1.ICE bit at 0 (SCLn and SDAn pins are in inactive state). This initializes the flag bits and internal state of the ICSR1 register. After that, set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2). Then, set the other resisters as necessary (see figure 35.5, Example of RIIC Initialization Flow, for the RIIC initialization procedure). When the necessary register settings have been completed, clear the ICCR1.IICRST bit to 0 (reset cancelation). This step is not necessary if the RIIC initialization has already been completed.

(2) Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 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 START flag in ICSR2 are automatically set to 1 and the ST bit is automatically cleared 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 the MST and TRS bits in ICCR2 are automatically set to 1, placing the RIIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, 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 cleared to 0 on the rising edge of the ninth cycle of SCLn (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the ICSR2.RDRF flag is automatically set to 1.
Since 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 ICDRR after confirming that the RDRF flag in ICSR2 is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.

(5) After 1 byte of data has been received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RDRFS bit in ICMR3. Reading out ICDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the ACKBT bit of ICMR3. 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 (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 rising 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 ACKBT bit in ICMR3 to 1 (NACK).

(7) After reading out 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 SP bit in ICCR2 (stop condition issuance request) and then read the last byte from ICDRR. When ICDRR 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 SCL line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears the MST and TRS bits in ICCR2 to “00b” and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the STOP flag in ICSR2 to 1.

(9) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

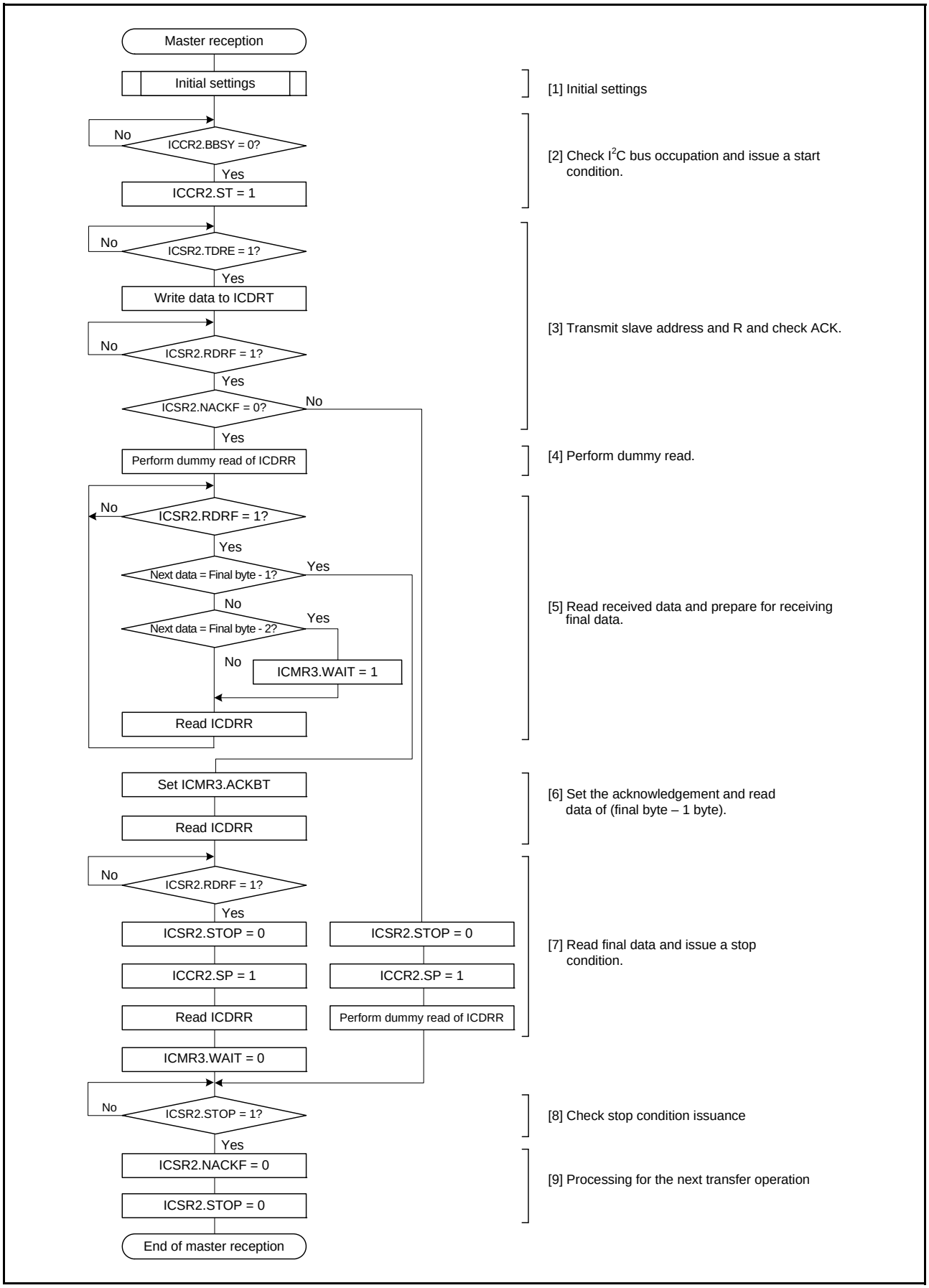


Figure 35.10 Example of Master Reception Flowchart (7-Bit Address Format)

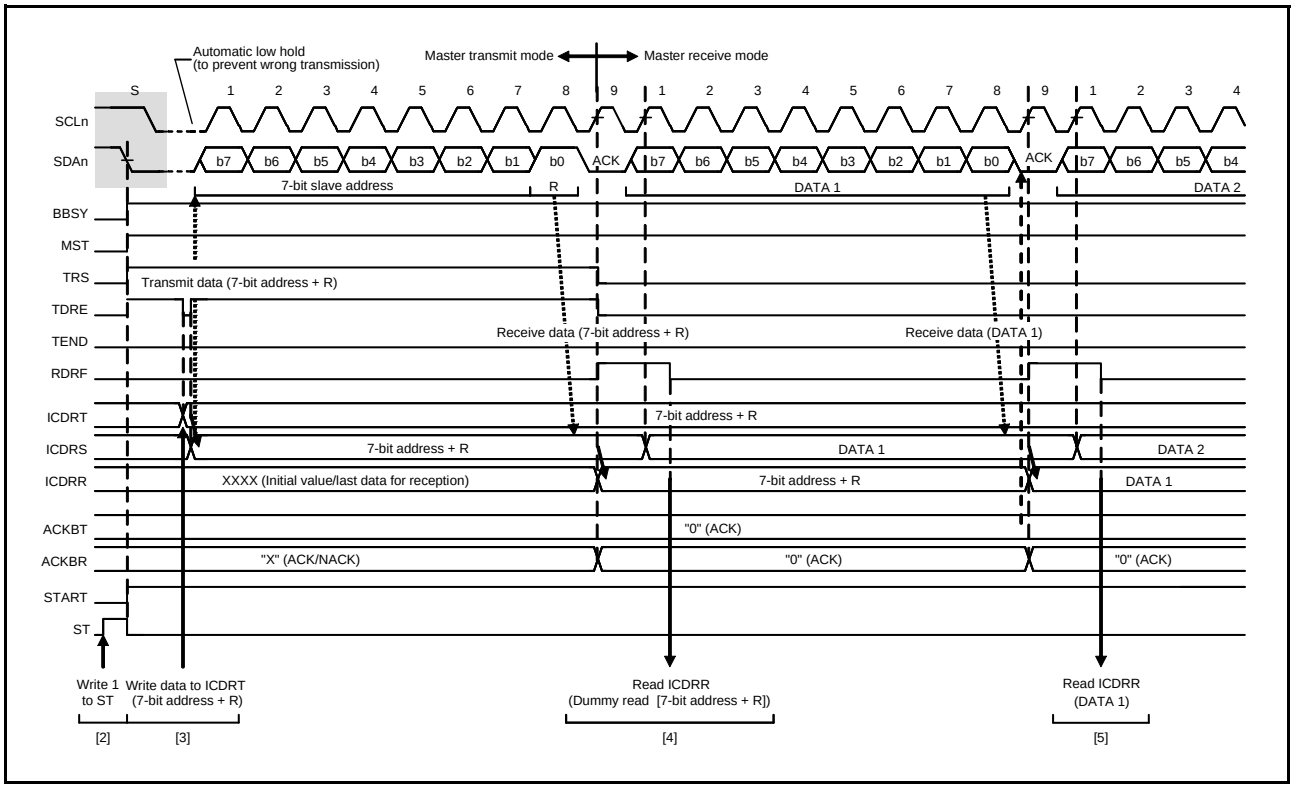


Figure 35.11 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

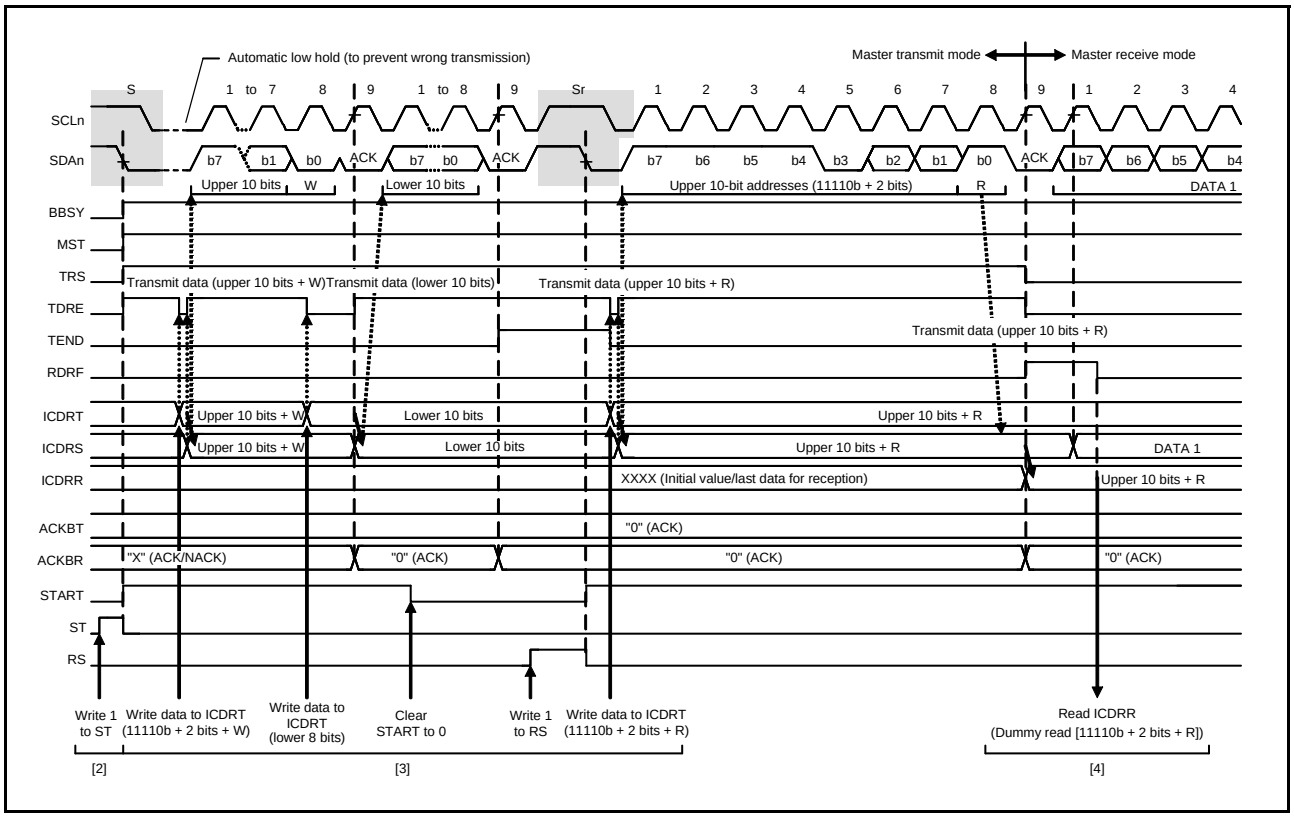


Figure 35.12 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

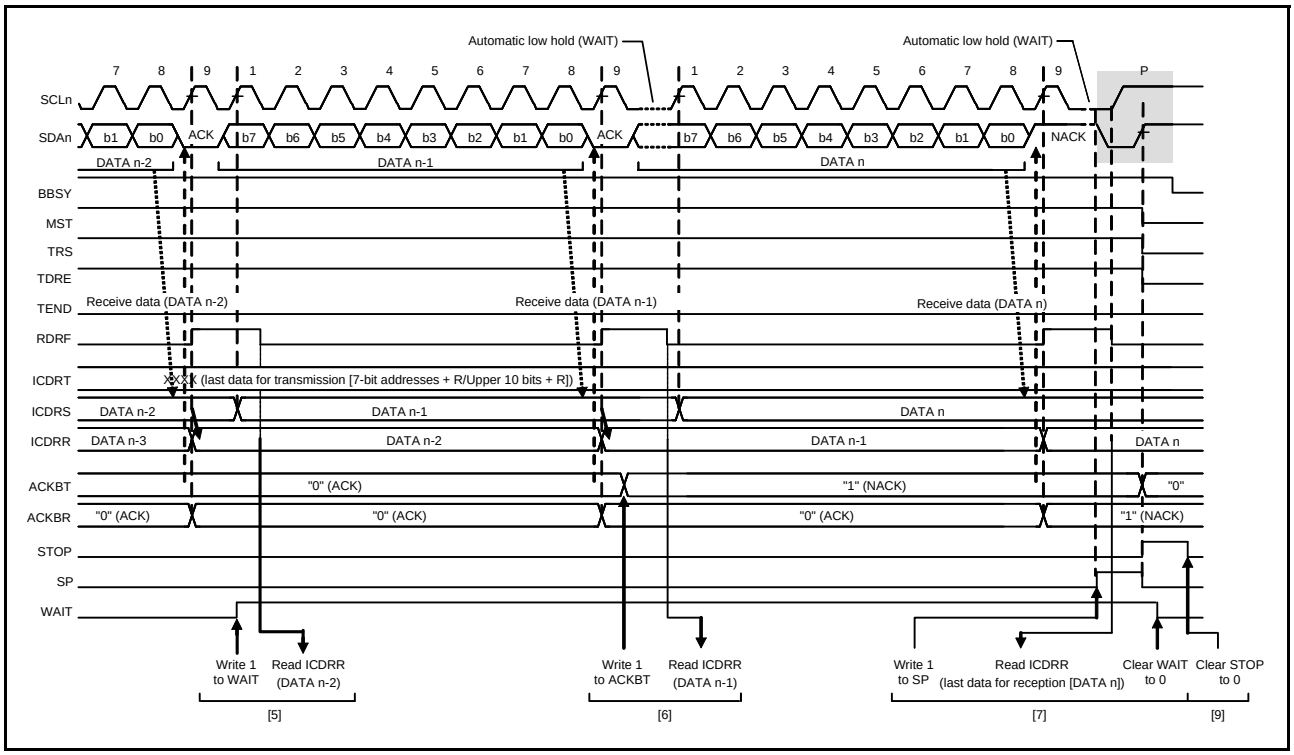


Figure 35.13 Master Receive Operation Timing (3) (when RDRFS = 0)

35.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 35.14 shows an example of usage of slave transmission and Figure 35.15 and Figure 35.16 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in Figure 35.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. 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 TRS bit and the TDRE flag in ICSR2 to 1.

(3) After the ICSR2.TEND 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 unit 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 ICDRR to complete the processing. This releases the SCLn line.

(6) Upon detecting the stop condition, the RIIC automatically clears 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, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

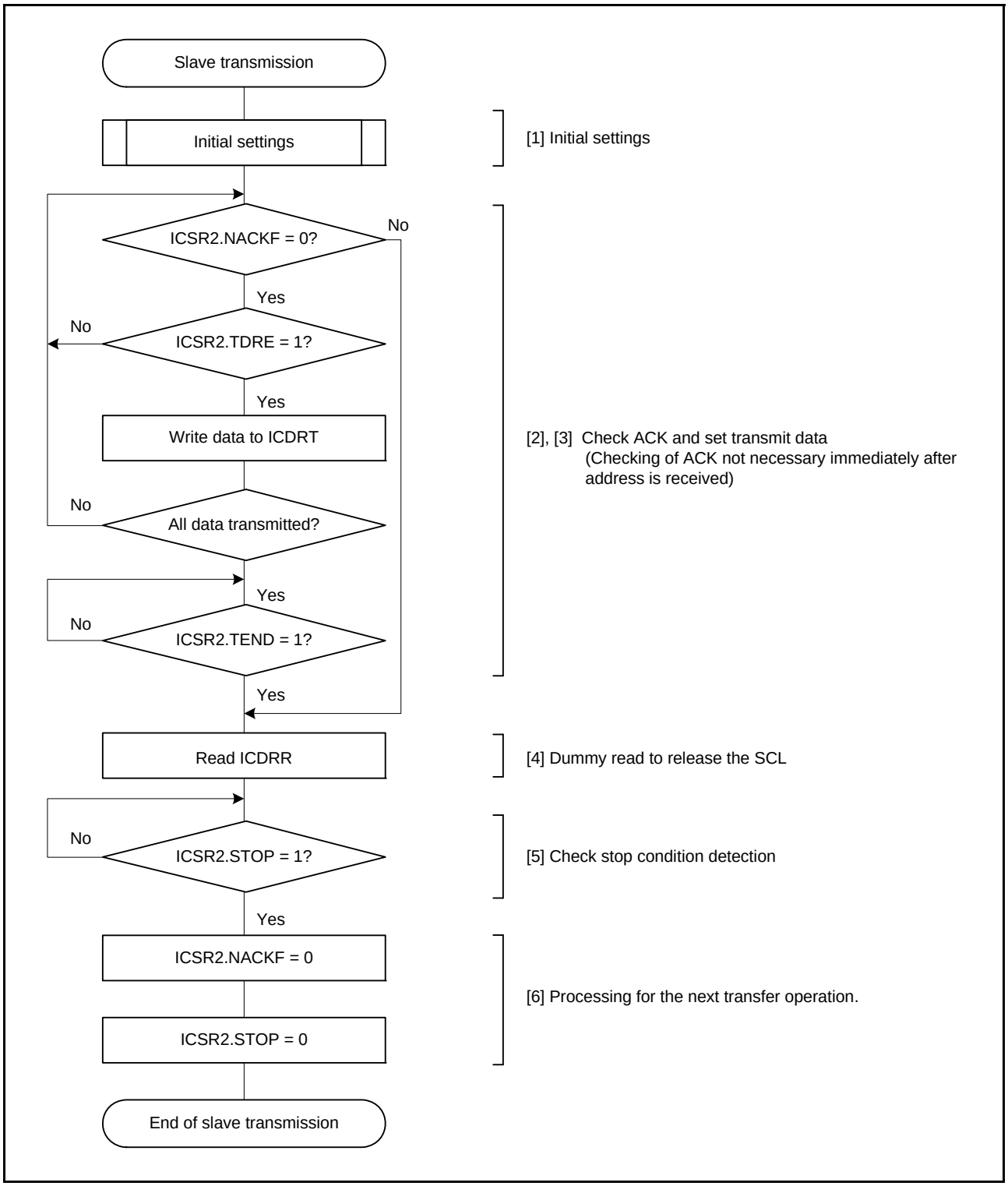


Figure 35.14 Example of Slave Transmission Flowchart

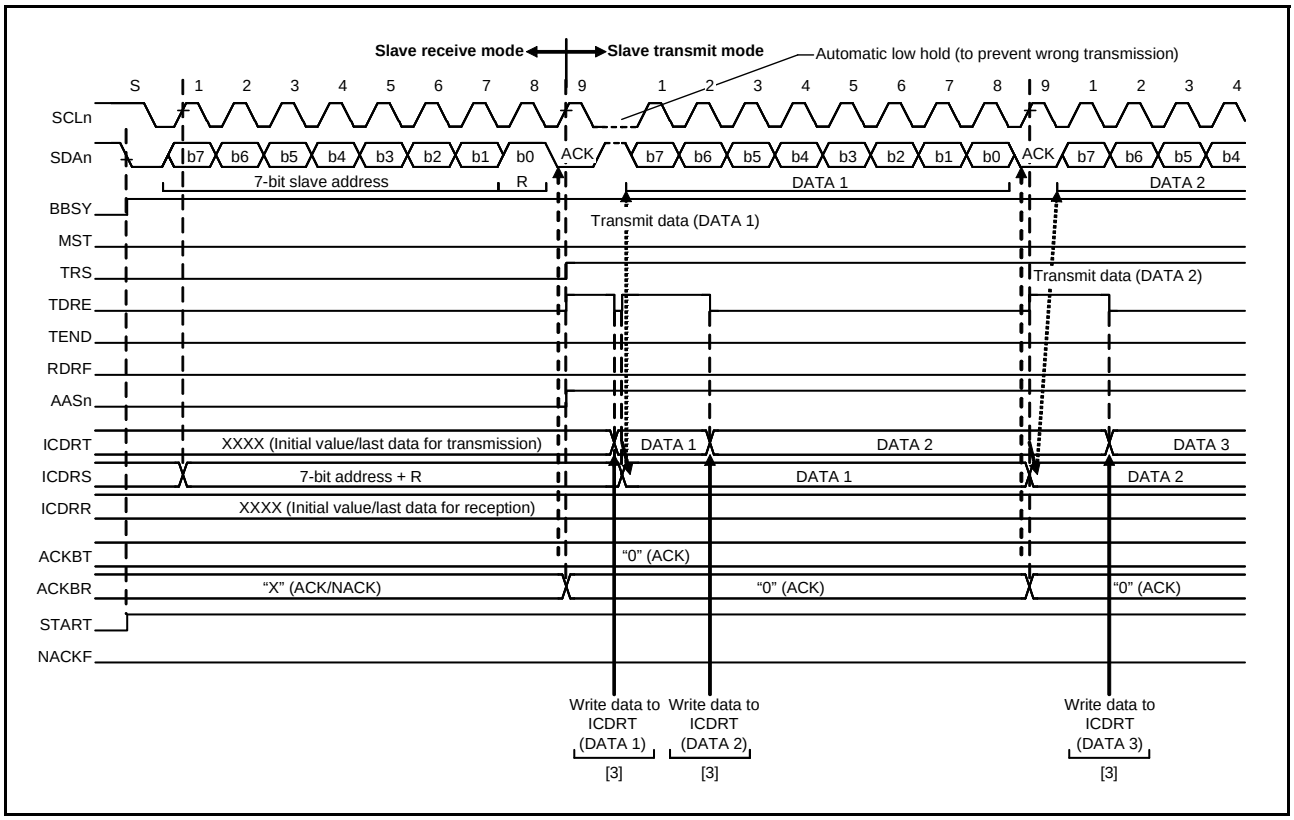


Figure 35.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

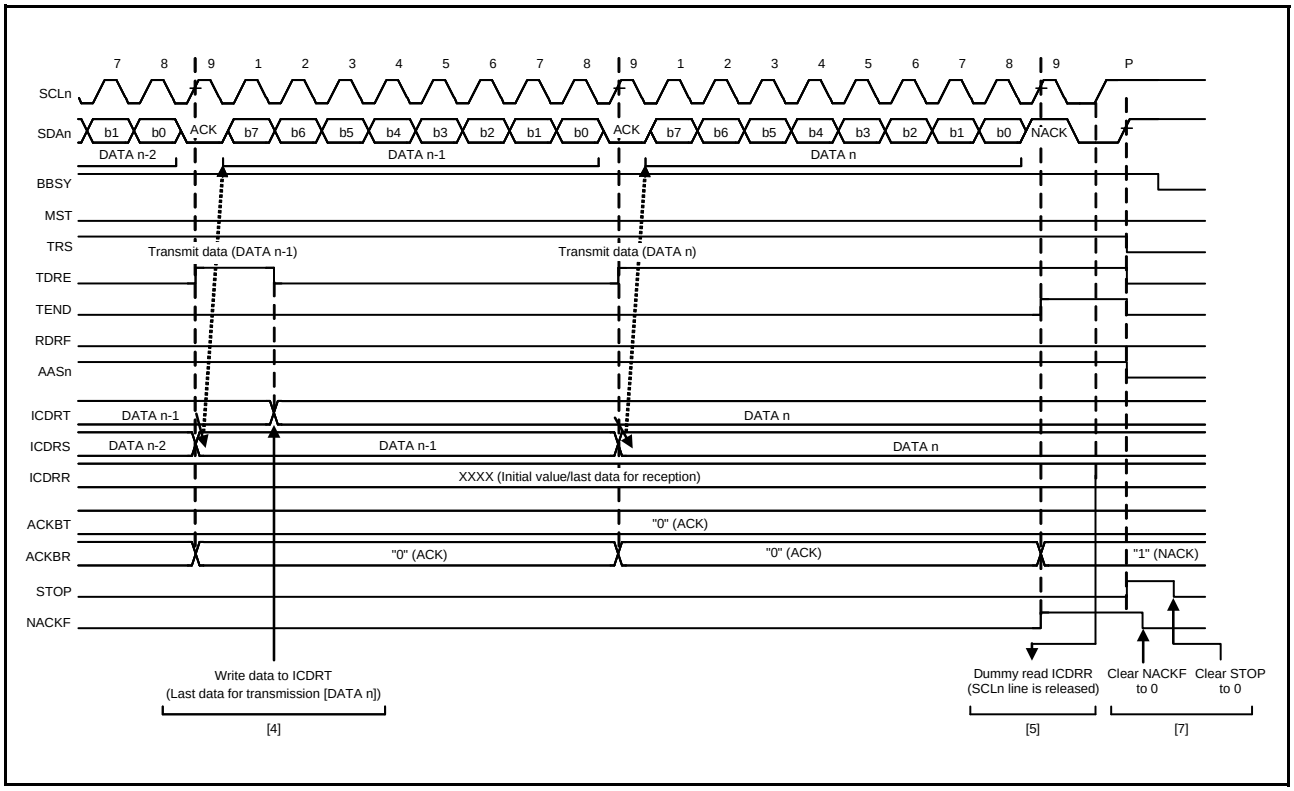


Figure 35.16 Slave Transmit Operation Timing (2)

35.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.
Figure 35.17 shows an example of usage of slave reception and Figure 35.18 and Figure 35.19 show the timing of operations in slave reception.
The following describes the procedure and operations for slave reception.

(1) Follow the procedure in Figure 35.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. 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 RDRF flag in ICSR2 to 1.
(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read ICDRR (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).
(4) When ICDRR is read, the RIIC automatically clears the ICSR2.RDRF flag to 0. If reading of ICDRR 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 ICDRR 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 ICDRR 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, clear the ICSR2. STOP flag to 0 for the next transfer operation.

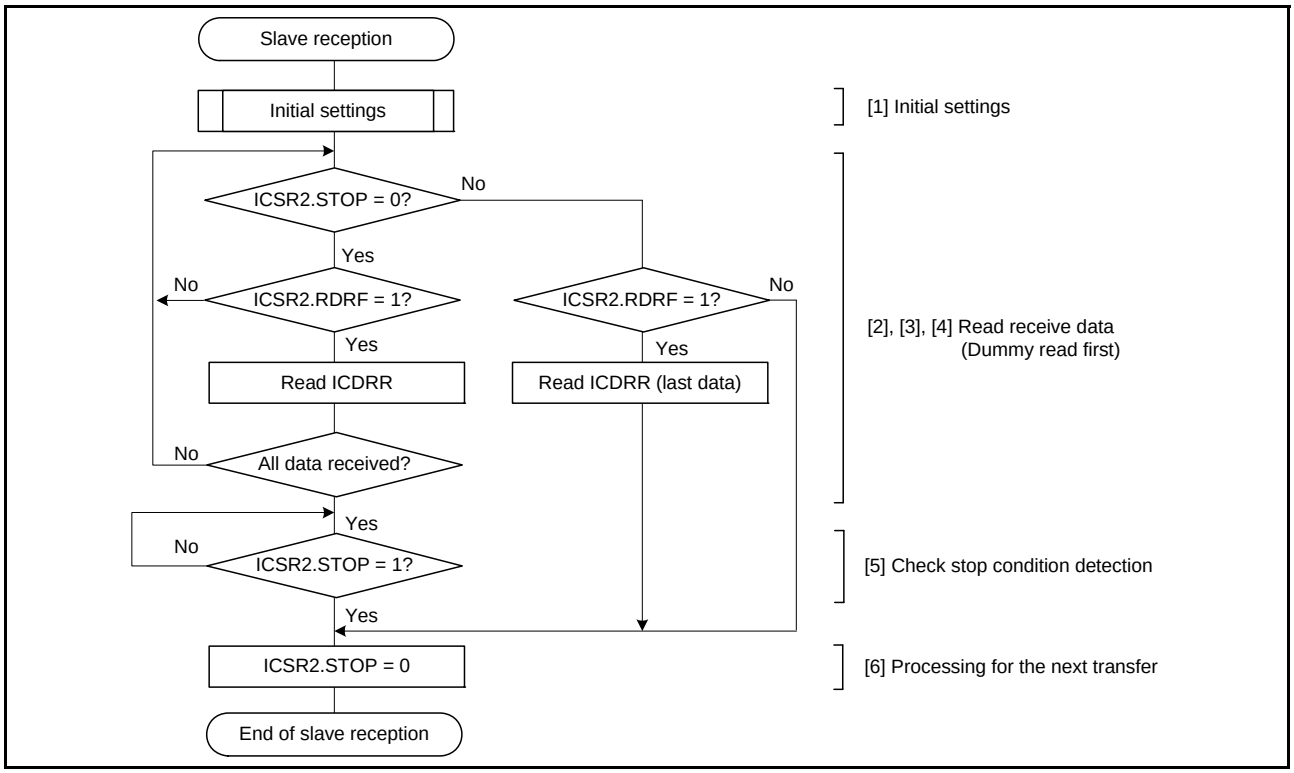


Figure 35.17 Example of Slave Reception Flowchart

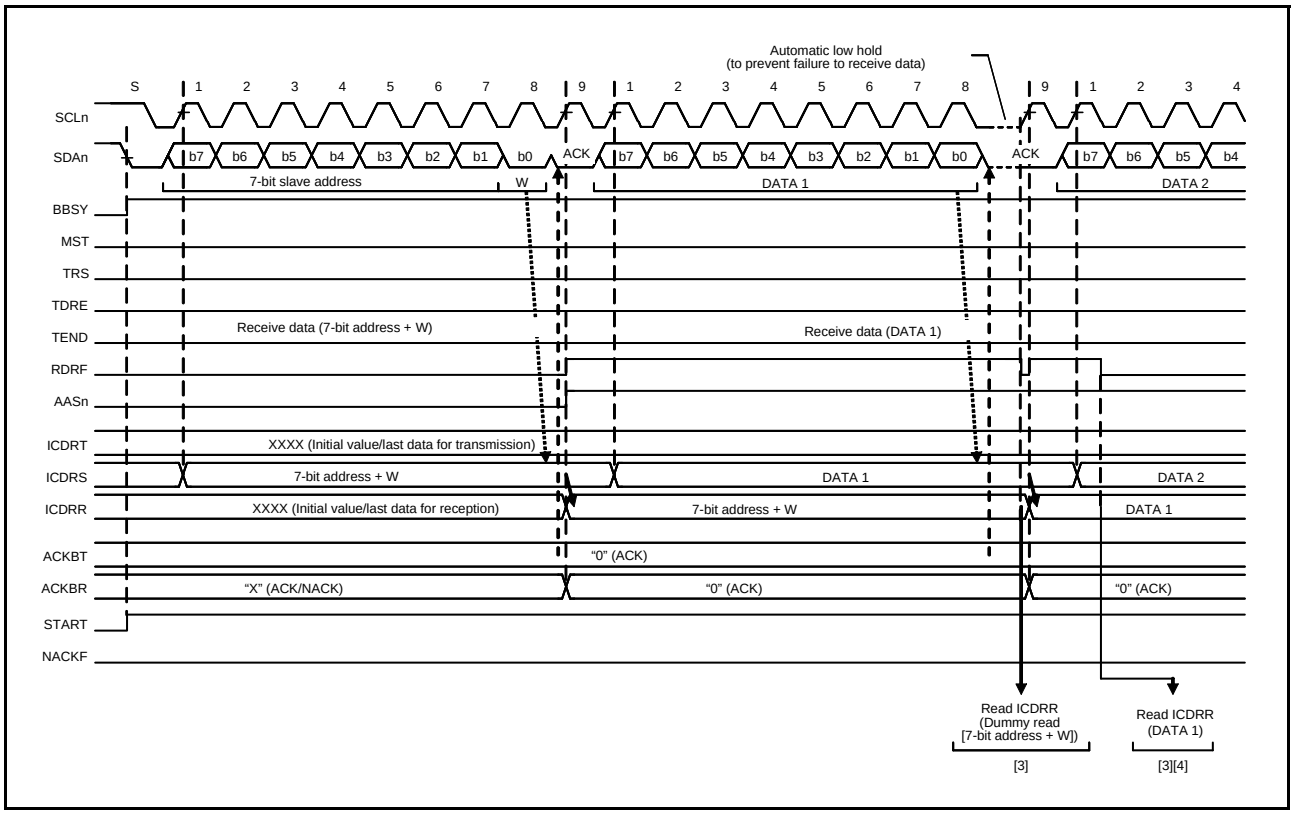


Figure 35.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

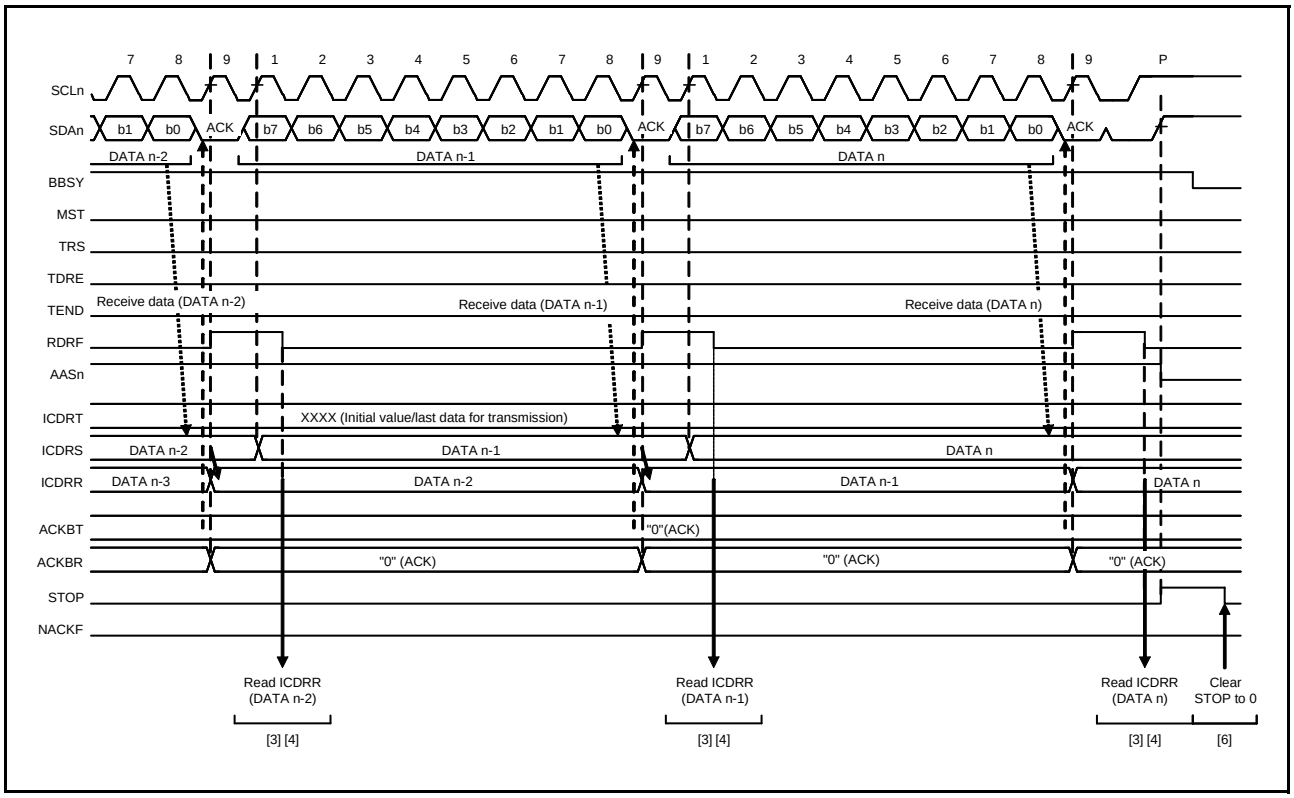


Figure 35.19 Slave Receive Operation Timing (2) (when RDRFS = 0)

35.4 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in ICBRH 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 ICBRL, and then stops driving the SCLn line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal 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. Since 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 ICBRH, 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 ICBRL. 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 SCL line has been released. When the RIIC finishes outputting the low-level period of the SCL clock of, 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 SCLE bit in ICFER is set to 1.

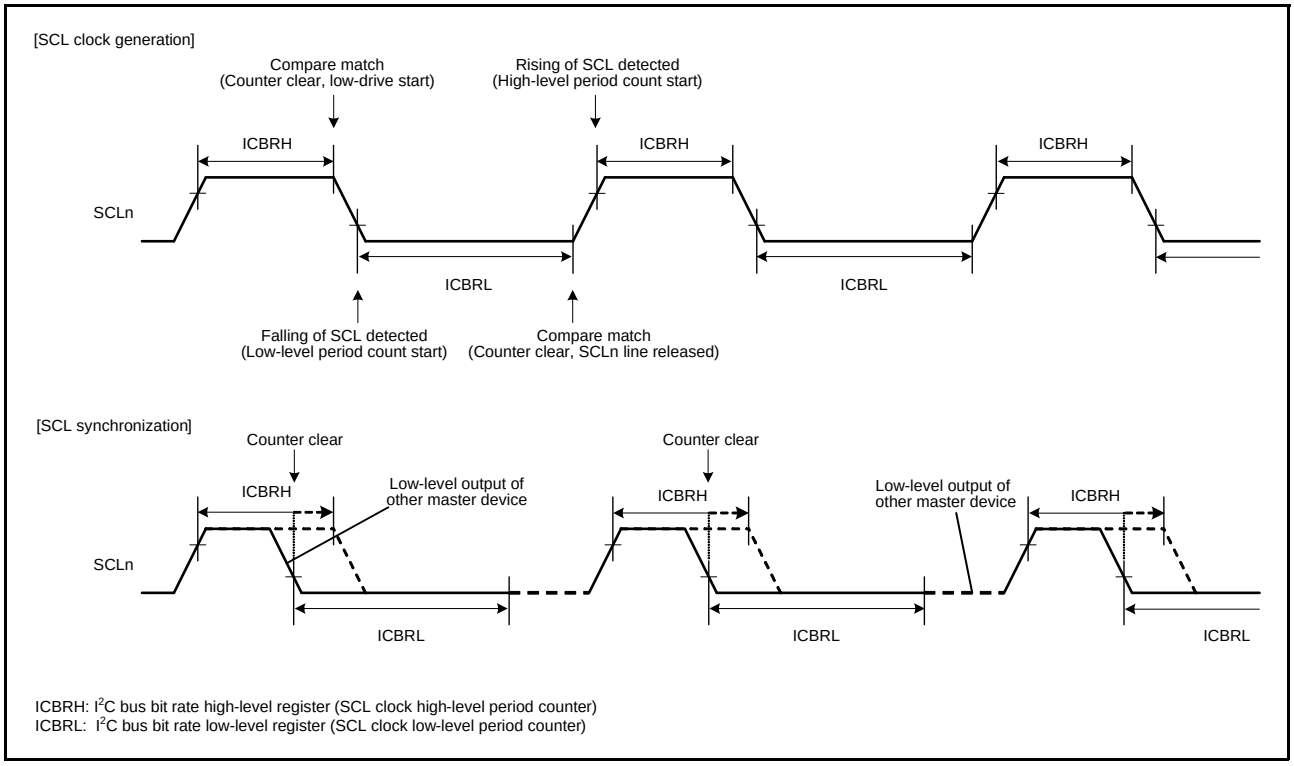


Figure 35.20 Generation and Synchronization of the SCL Signal from the RIIC

35.5 Facility for Delaying SDA Output

The RIIC module incorporates a facility 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 facility, 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) signal 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 facility is enabled by setting the SDDL[2:0] bits in IMCR2 to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits in IMCR2 are set to any value other than 000b), the DLCS bit in IMCR2 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 SDDL[2:0] bits in IMCR2. 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.

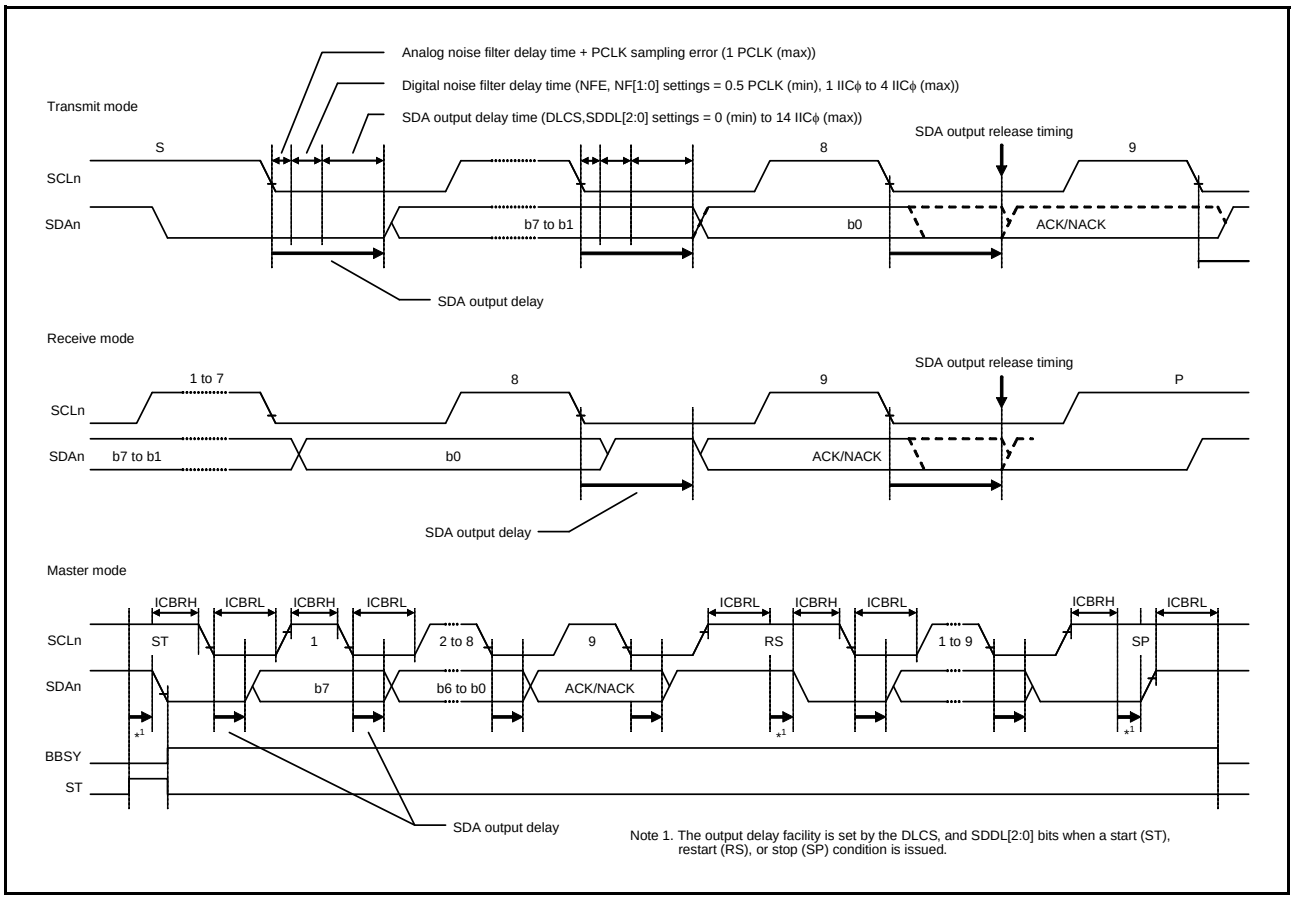


Figure 35.21 SDA Output Delay Facility

35.6 Digital Noise-Filter Circuits

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 35.22 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 NF[1:0] bits in ICMR3. 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 NF[1:0] bits in ICMR3, 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 clearing the NFE bit in ICFER) and use only the analog noise-filter circuit.

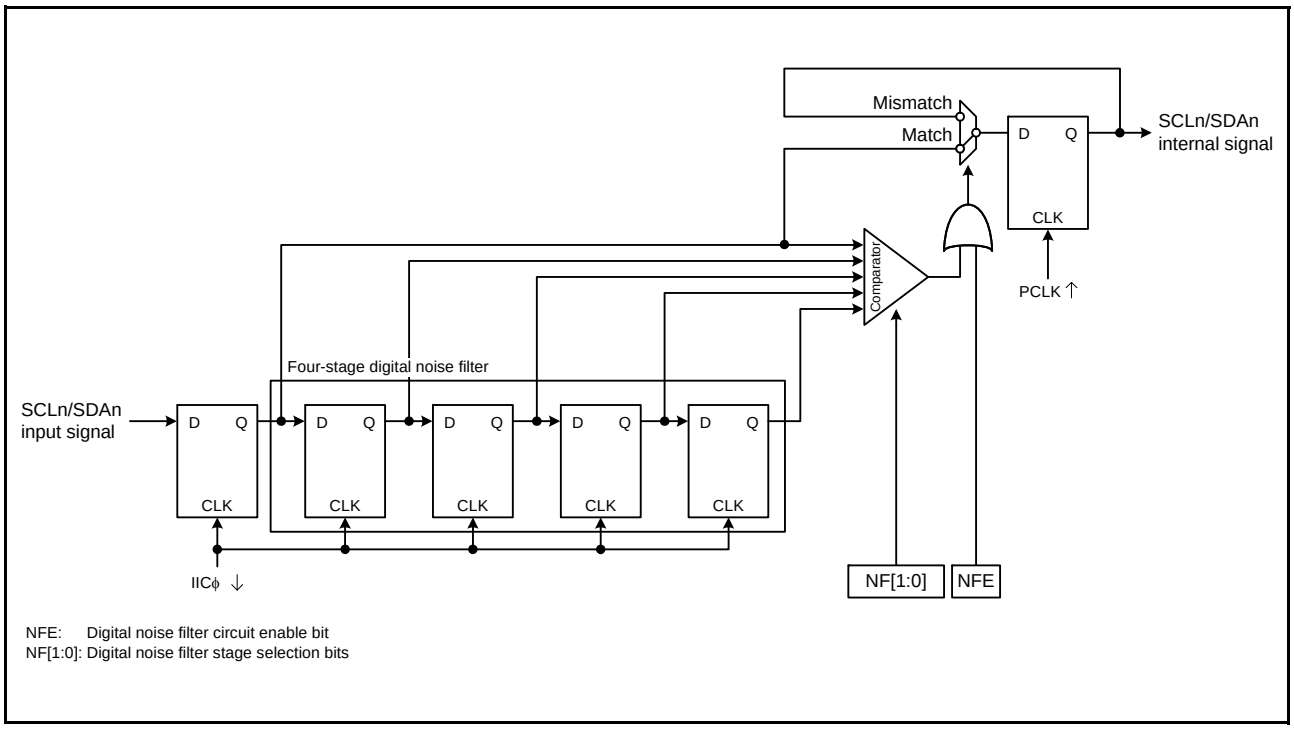


Figure 35.22 Block Diagram of Digital Noise Filter Circuit

35.7 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.

35.7.1 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 SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding AASy flag (y = 0 to 2) in ICSR1 is set to 1 at the falling edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the following R/W# bit. This causes a receive data full interrupt (ICRXI) or transmit data empty interrupt (ICTXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 35.23 to Figure 35.25 show the AASy flag set timing in three cases.

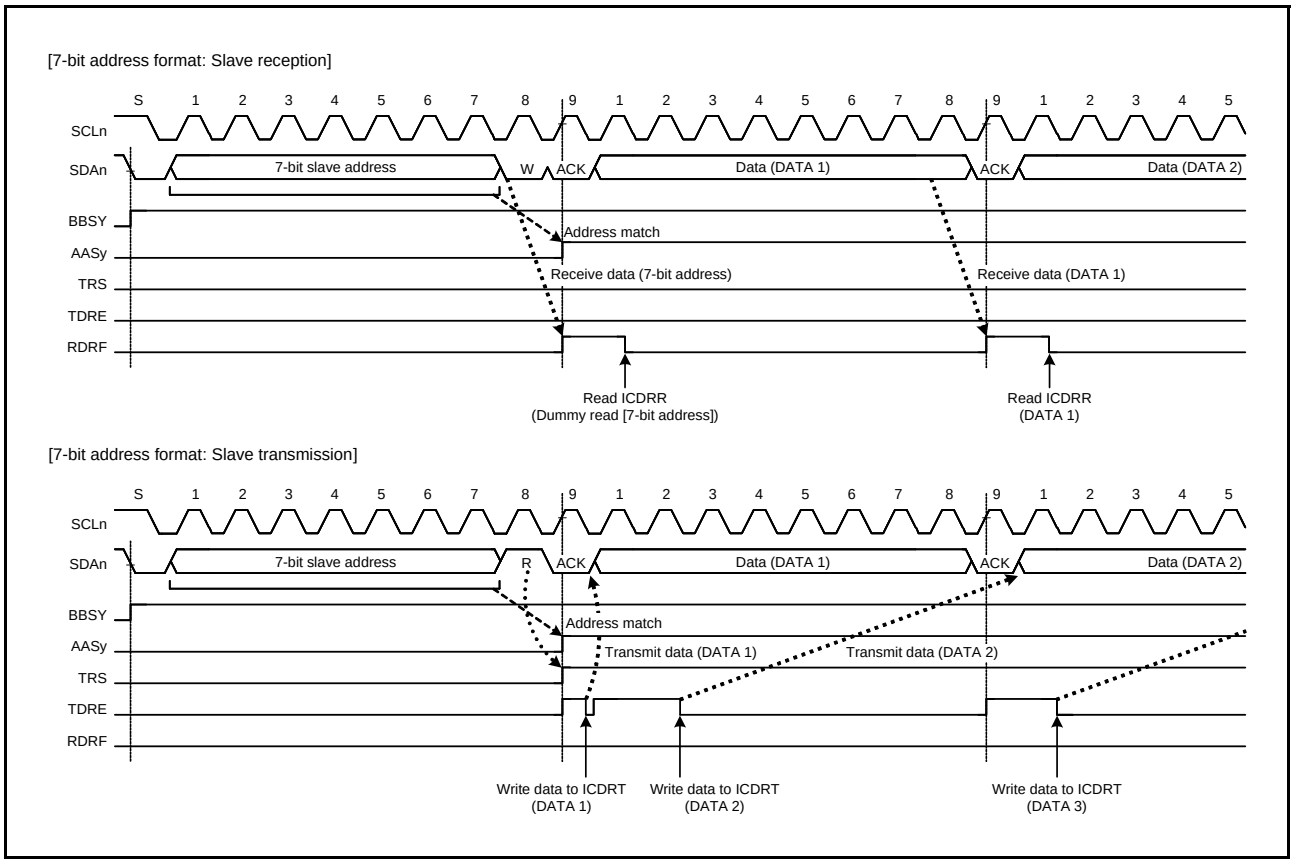


Figure 35.23 AASy Flag Set Timing with 7-Bit Address Format Selected

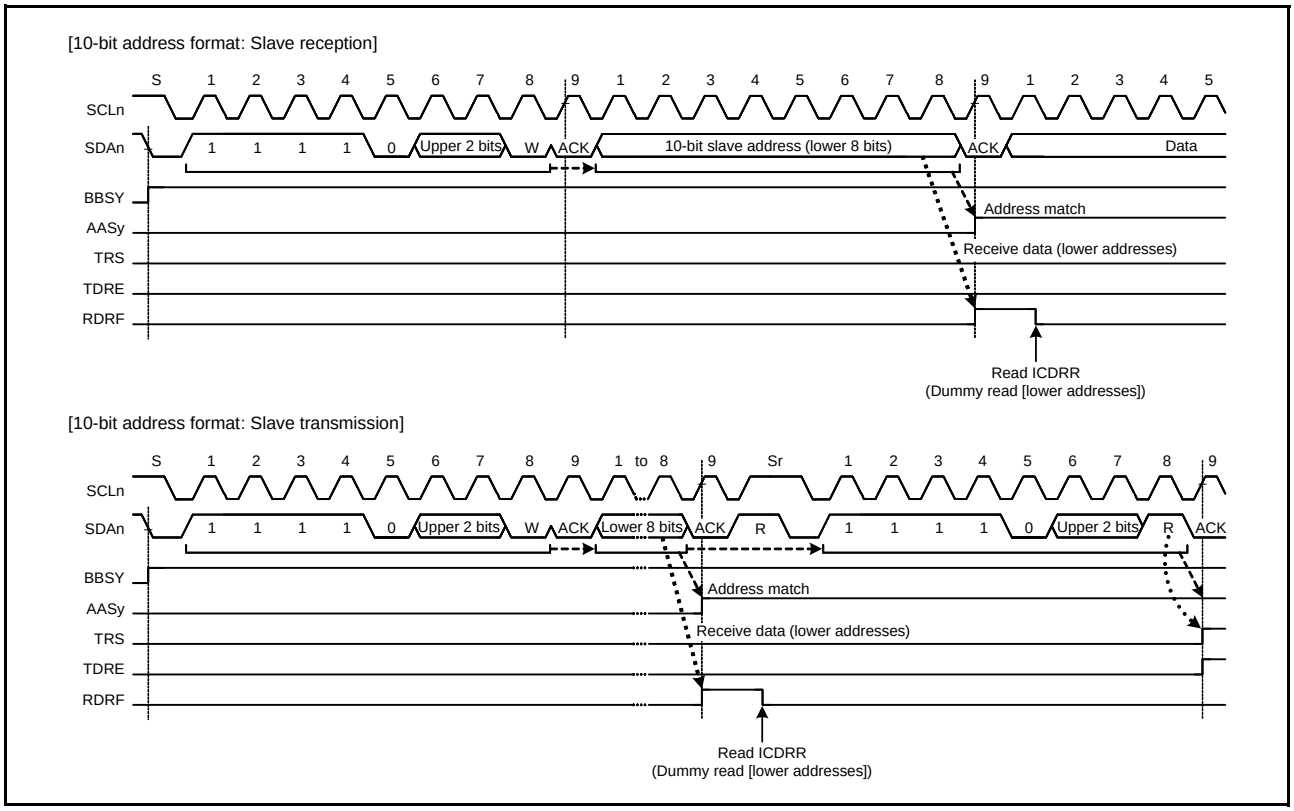


Figure 35.24 AASy Flag Set Timing with 10-Bit Address Format Selected

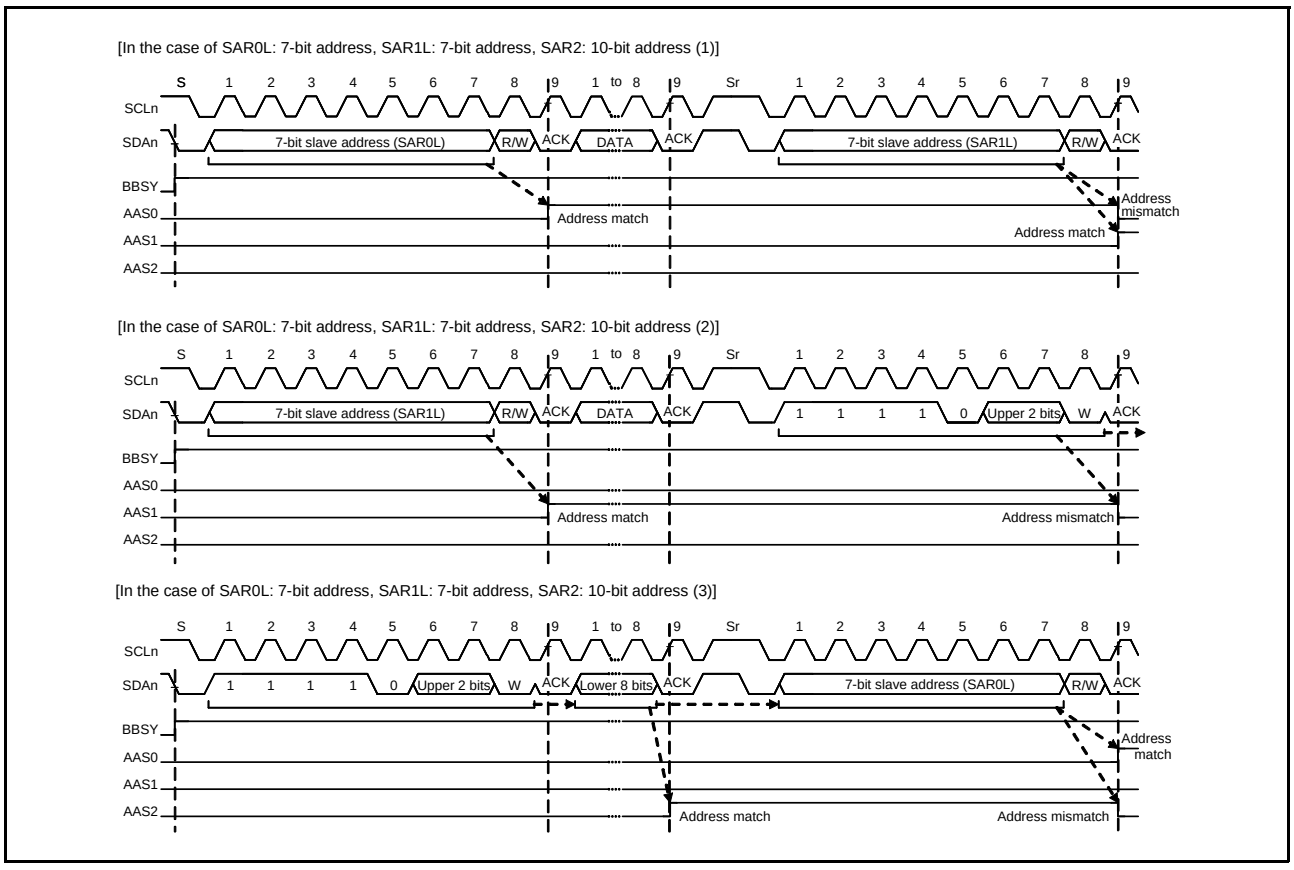


Figure 35.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

35.7.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (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 GCA flag in ICSR1 and the RDRF flag in ICSR2 are set to 1 on the falling edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (ICRXI). 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.

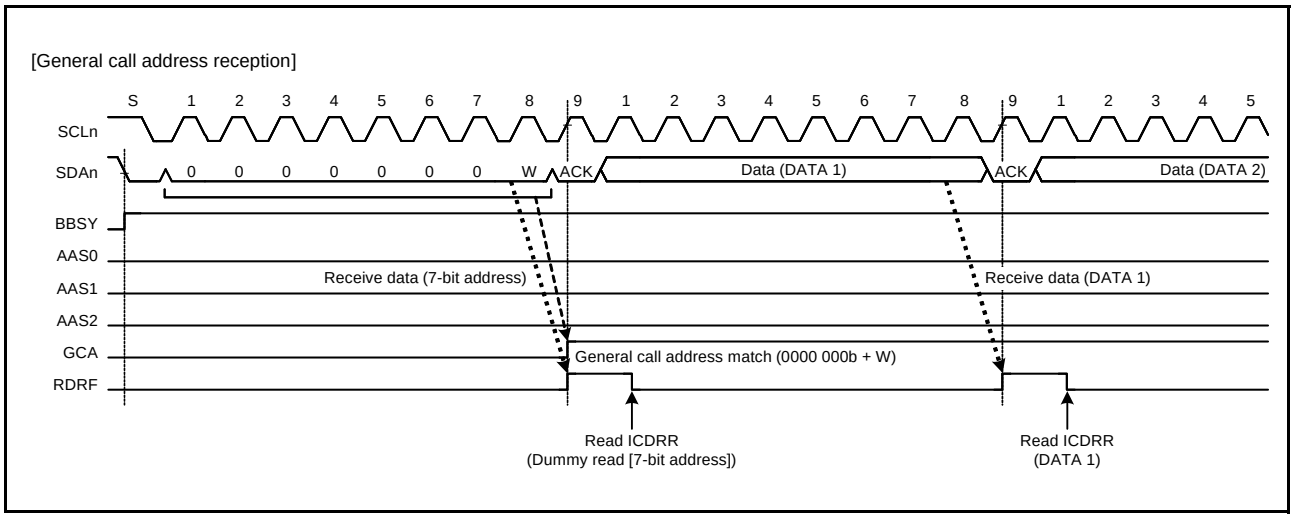


Figure 35.26 Timing of GCA Flag Setting during Reception of General Call Address

35.7.3 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 DIDE bit in ICSER set to 1, the RIIC recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the ninth 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 AASy flag (y = 0 to 2) in ICSR1 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 clears 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 = 1.

Furthermore, prepare the device-ID fields (three 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.

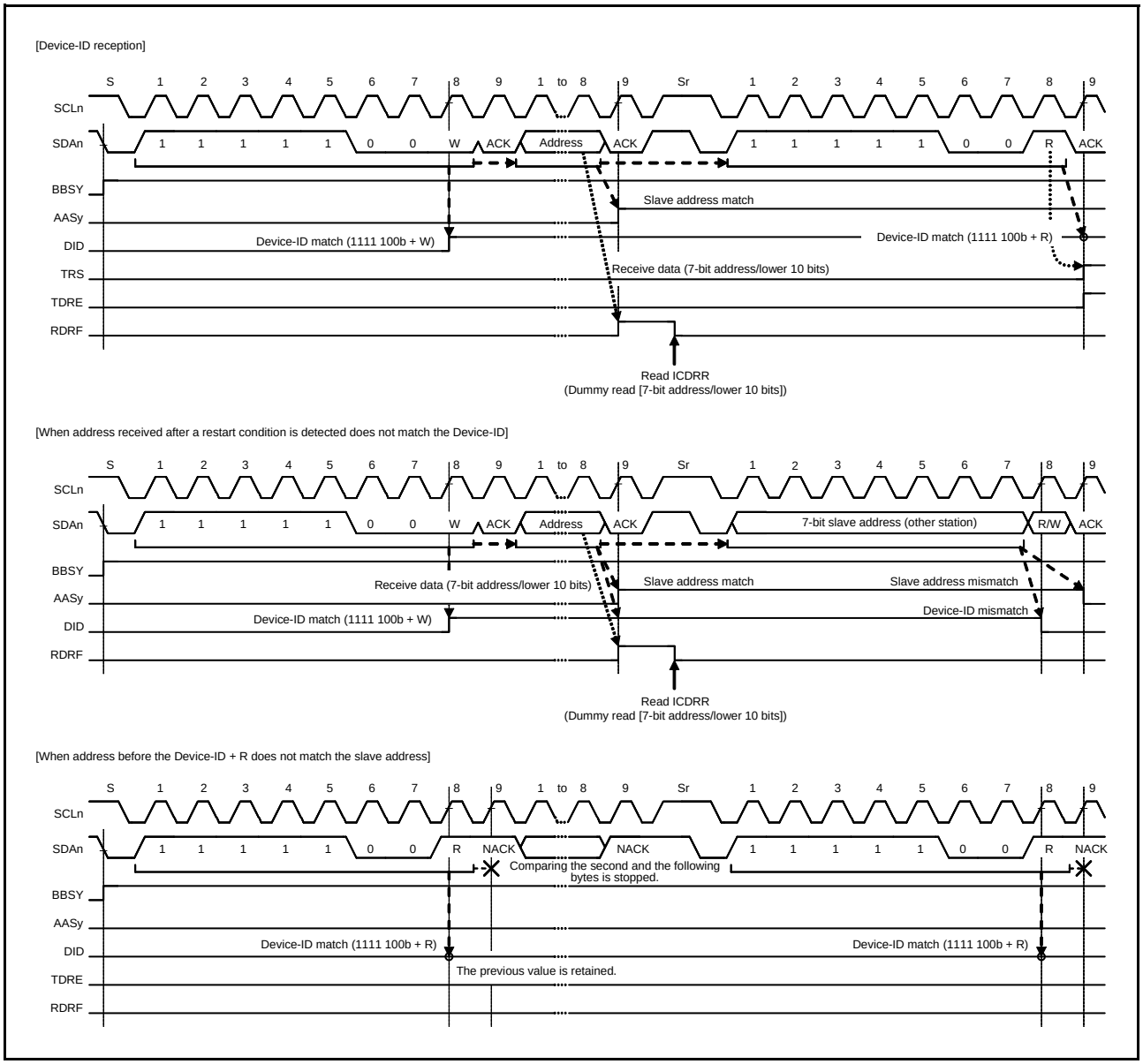


Figure 35.27 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

35.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the RIIC detects the host address, the HOA flag in ICSR1 is set to 1 at the falling edge of the ninth SCL clock cycle, and at the same time, the TDRE flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a transmit data empty interrupt (ICTXI) 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 = 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.

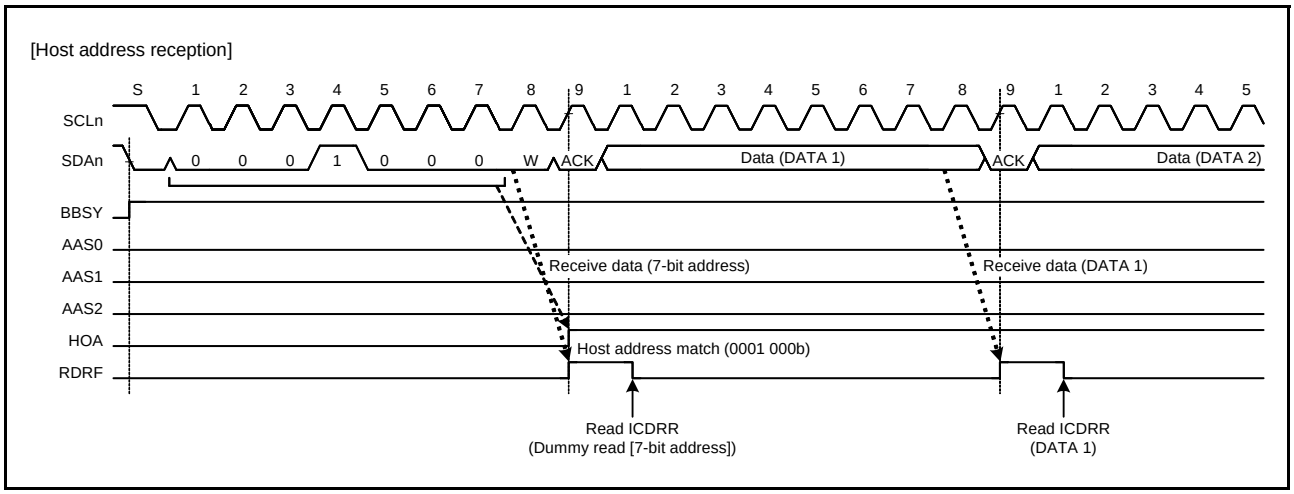


Figure 35.28 HOA Flag Set Timing during Reception of Host Address

35.8 Automatically Low-Hold Function for SCL

35.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the transmit data register (ICDRT) with the RIIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn signal 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 of one clock cycle 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

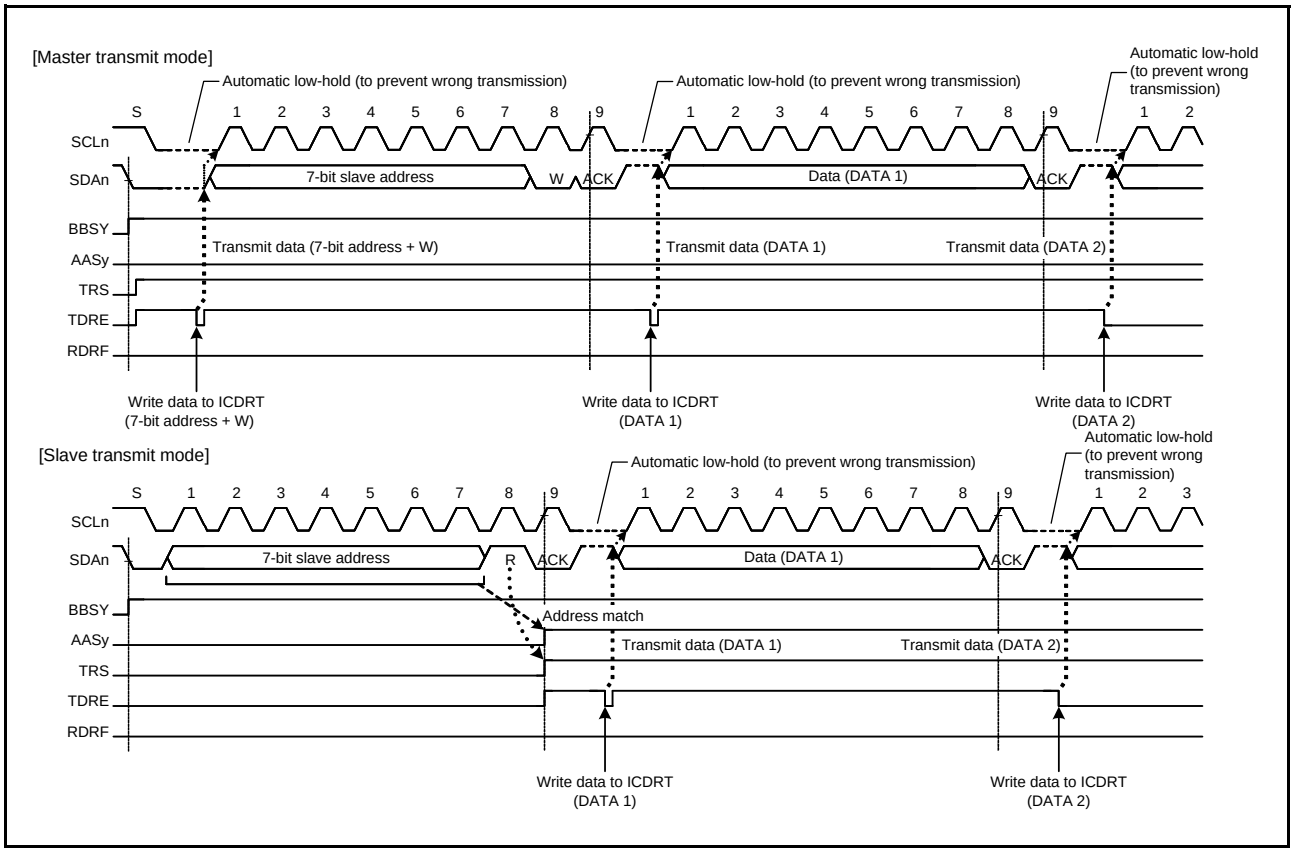


Figure 35.29 Automatic Low-Hold Operation in Transmit Mode

35.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1 (transfer suspension enabled). If the next transmit data has already been written (TDRE flag = 0 in ICSR2) 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 (NACKF flag = 1 in ICSR2), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

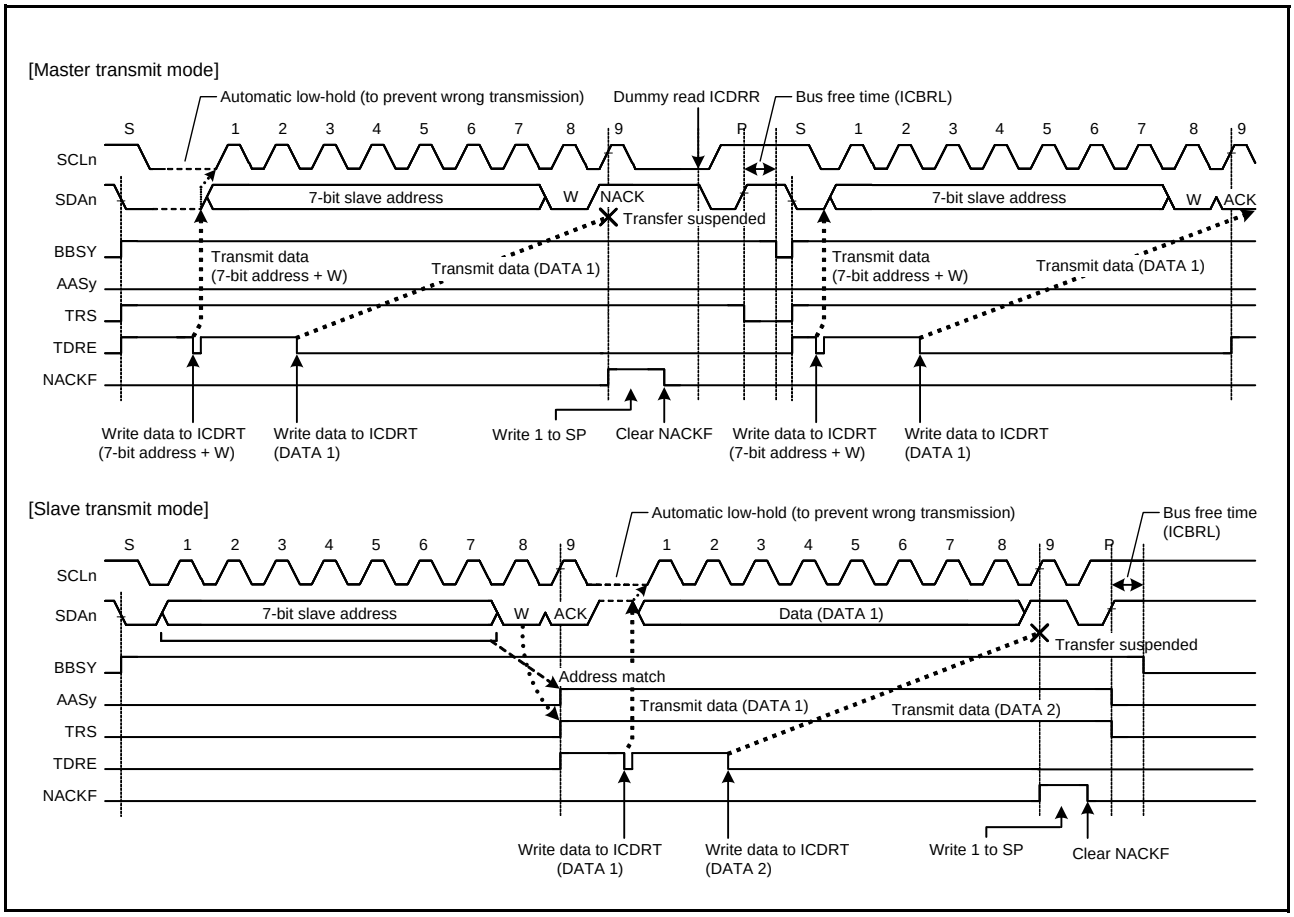


Figure 35.30 Suspension of Data Transfer when NACK is Received (NACKE = 1)

35.8.3 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 frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), 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 ICMR3.

(1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the WAIT bit in ICMR3 is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ACKBT bit value in ICMR3 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 ICDRR, which enables bytewise receive operation.

The WAIT bit function is enabled for receive frames 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) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RDRFS bit in ICMR3 is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag (receive data full) in ICSR2 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 ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, 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 frames 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.

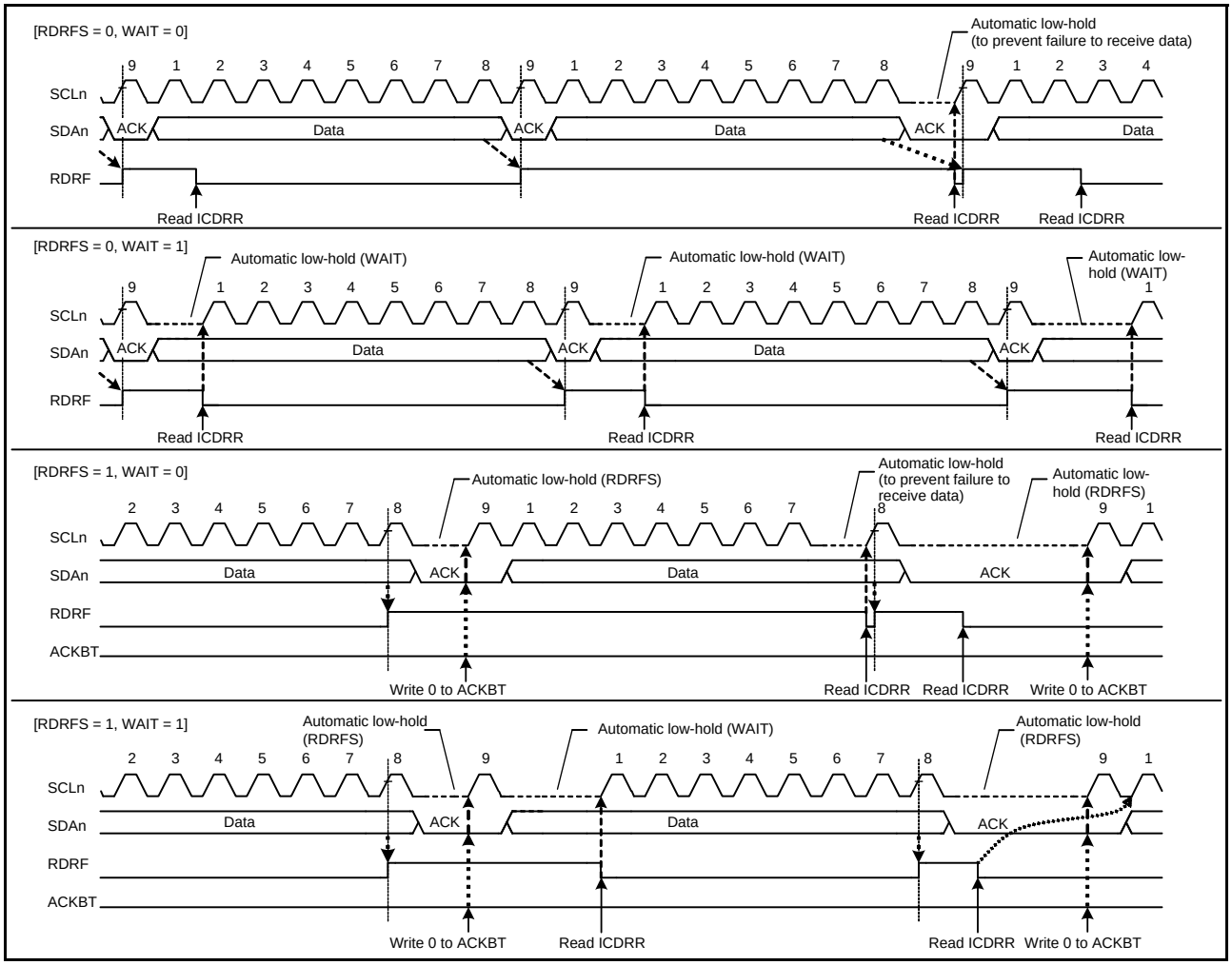


Figure 35.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

35.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C bus standard, 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.

35.9.1 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 regards its own issuing of a start condition as an error and considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the RIIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration, thus preventing a failure of transfer due to issuing of a start condition while transfer is in progress.
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 1 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 SDA 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 MALE bit in ICFER is 1 (master arbitration-lost detection enabled).
[Master arbitration-lost conditions]

- 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 ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 was cleared to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 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 (MST and TRS bits = 11b in ICCR2)

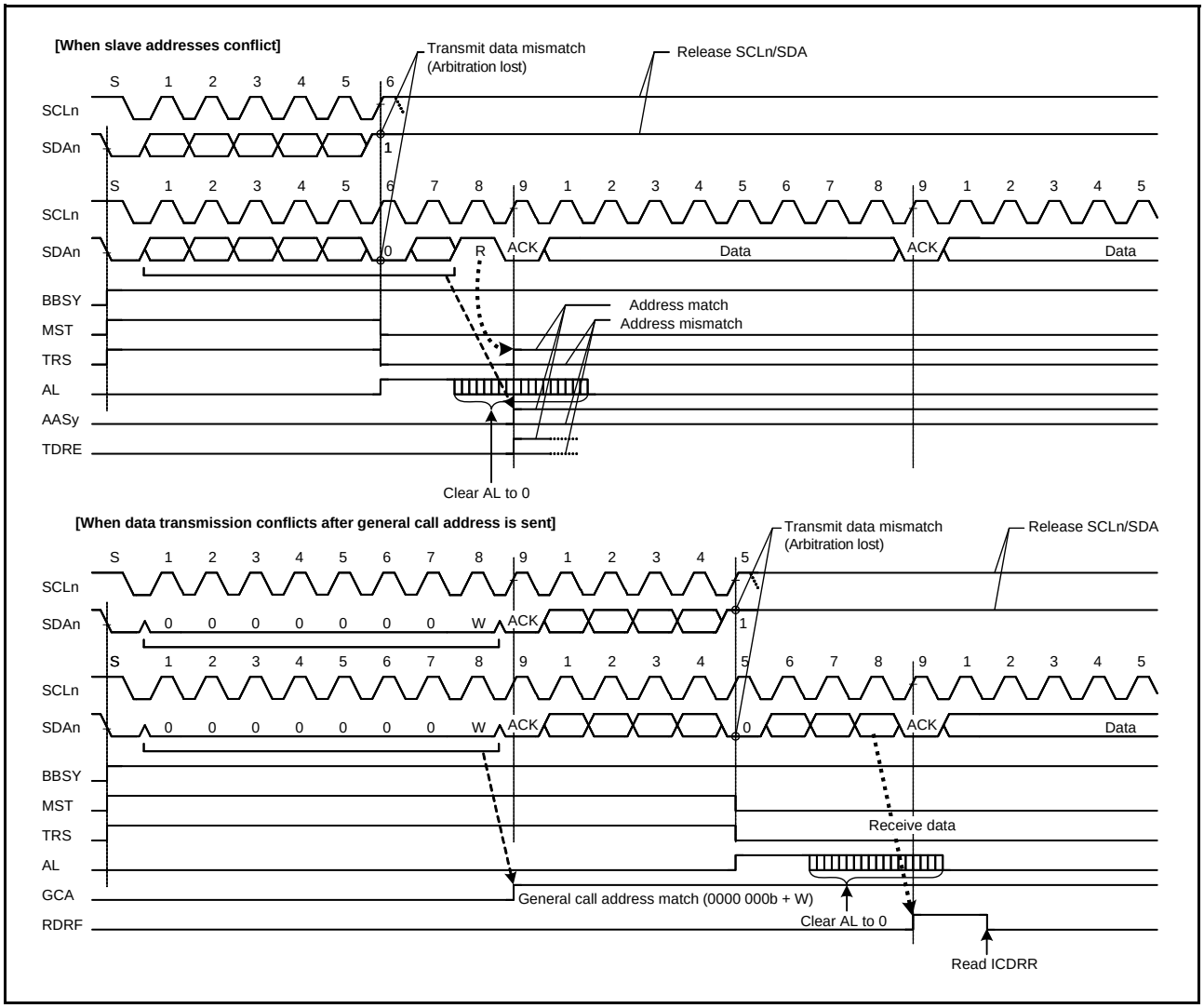


Figure 35.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

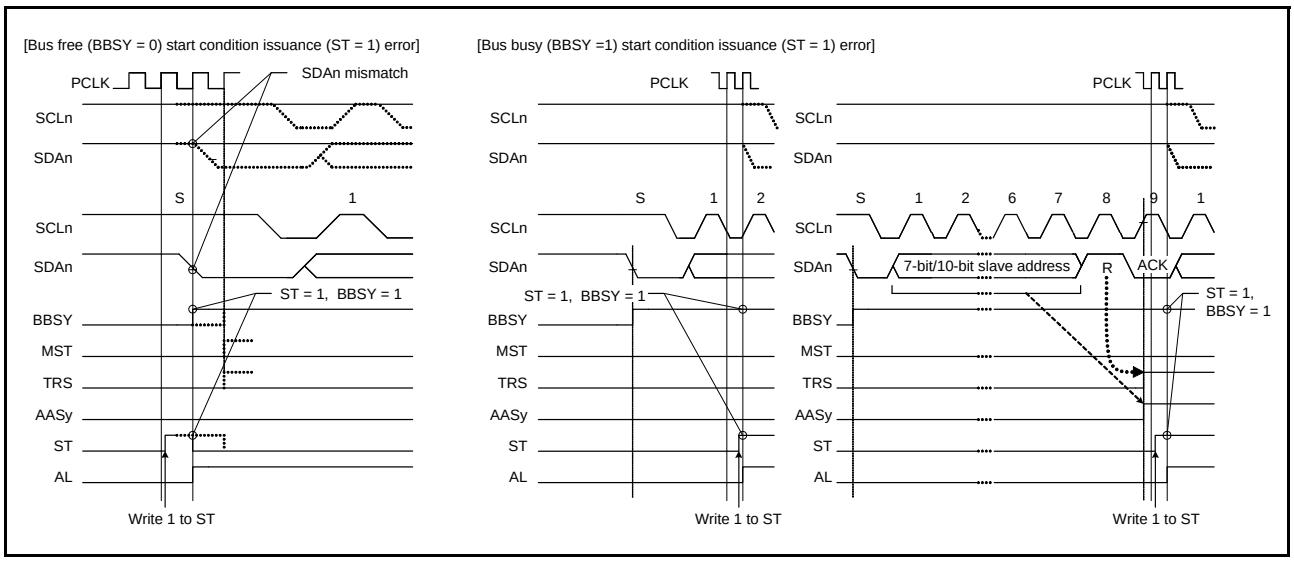


Figure 35.33 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

35.9.2 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 1 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 SDA 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 35.34 shows an example of arbitration-lost detection during transmission of NACK.

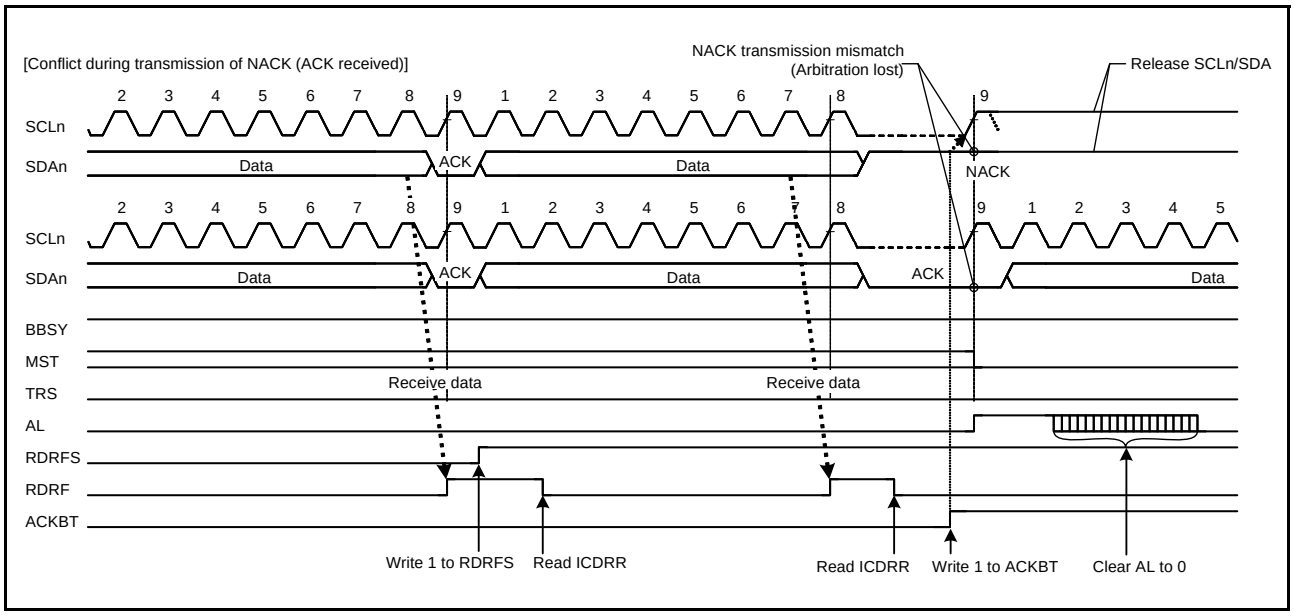


Figure 35.34 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

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 two bytes of data from the slave device, and master B receives four 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 two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four 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 dose 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 NALE bit in ICFER 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 (ACKBT bit = 1 in ICMR3)

35.9.3 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 1 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 SDA 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 eliminates subsequent redundant processing (processing for the transmission of FFh).
The RIIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER 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 (MST and TRS bits = 01b in ICCR2)

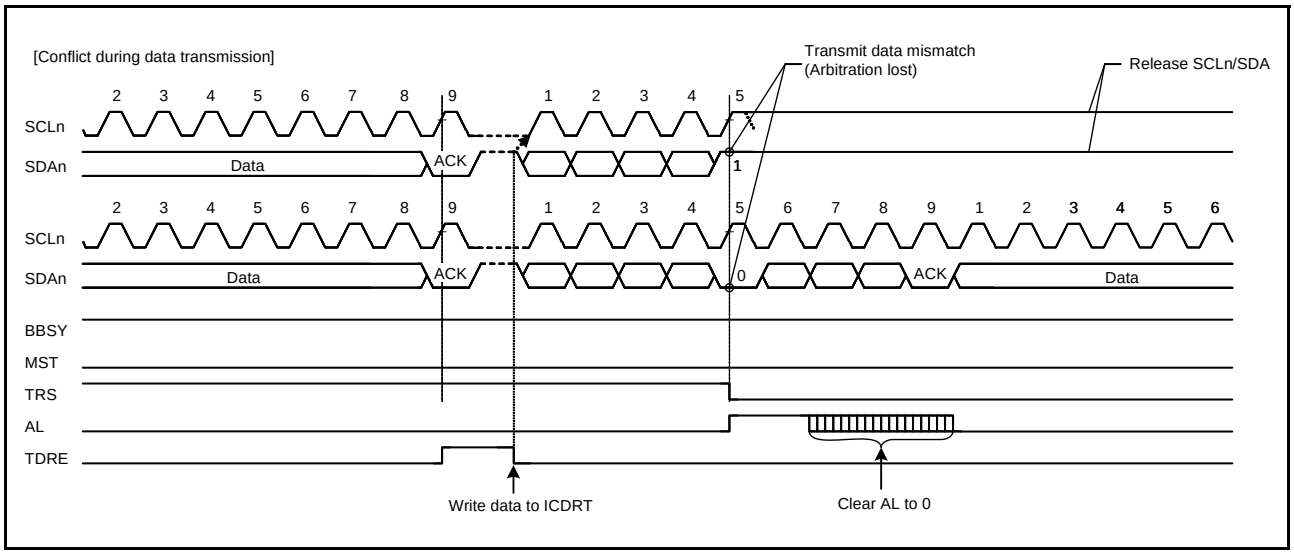


Figure 35.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

35.10 Start Condition/Restart Condition/Stop Condition Issuing Function

35.10.1 Issuing a Start Condition

The RIIC issues a start condition when the ST bit in ICCR2 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 BBSY flag in ICCR2 is 0 (bus free). 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 ICBRH 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 ICBRL.

35.10.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RS bit in ICCR2 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 BBSY flag in ICCR2 is 1 (bus busy) and the MST bit in ICCR2 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 ICBRL.
(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 ICBRL and the restart condition setup time.
(5) Drive the SDAn line low (high level to low level).
(6) Ensure the time set in ICBRH 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 ICBRL.

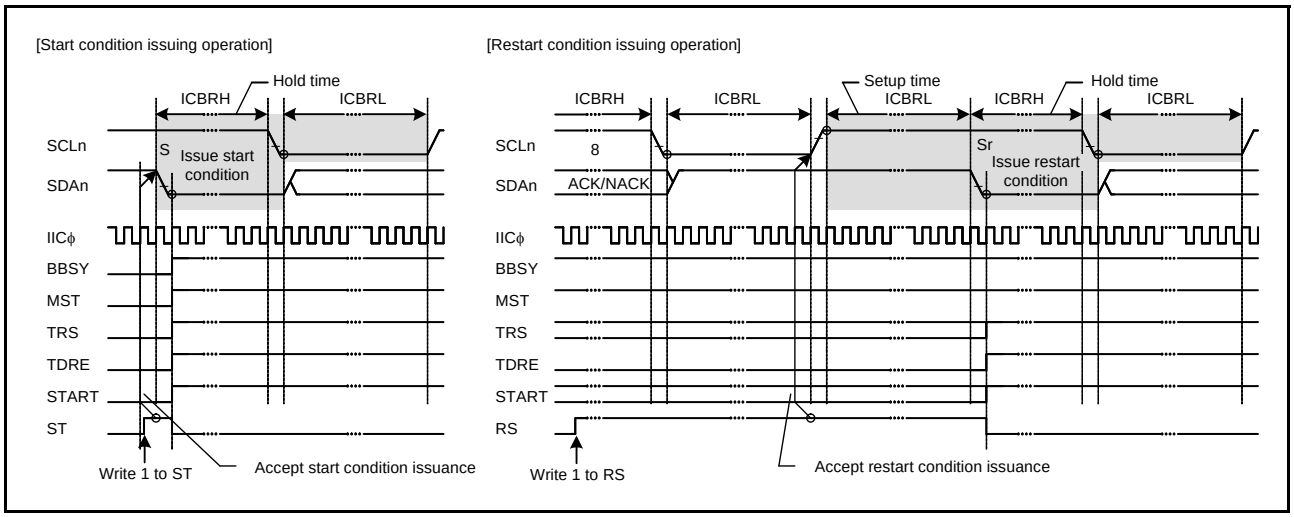


Figure 35.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

35.10.3 Issuing a Stop Condition

The RIIC issues a stop condition when the SP bit in ICCR2 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 BBSY flag in ICCR2 is 1 (bus busy) and the MST bit in ICCR2 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 ICBRL.
- Release the SCLn line (low level to high level).
- Detect a high level of the SCLn line and ensure the time set in ICBRH and the stop condition setup time.
- Release the SDAn line (low level to high level).
- Ensure the time set in ICBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

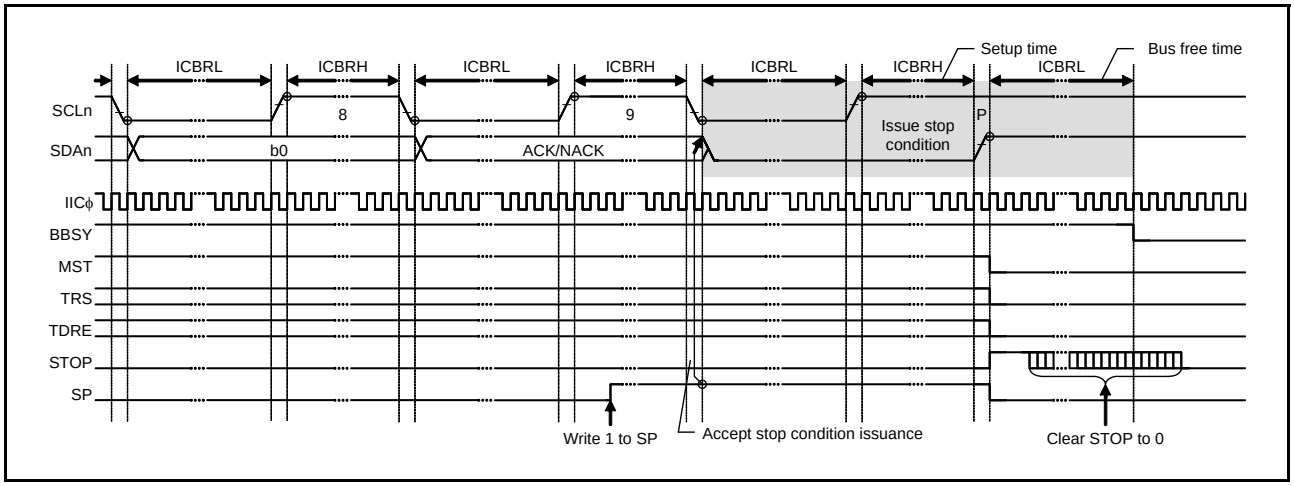


Figure 35.37 Stop Condition Issue Timing (SP Bit)

35.11 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, and the RIIC/internal reset function.
By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCLn or SDAn lines.

35.11.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCLn line is held for a certain period of time. In the bus busy state, the RIIC can detect an abnormal bus state by monitoring that the SCLn line is held 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 abnormality.

This timeout function is enabled when the TMOE bit in ICFER is 1. It detects an abnormal bus state that the SCLn line is held low or high when the bus is busy (BBSY flag = 1 in ICCR2) in master mode or when the BBSY flag is 1 and the RIIC's own slave address matches (ICSR1 is not 00h) in slave mode.

The internal counter of the timeout function works using the internal reference clock (IICϕ) set by the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low/high or both levels) during which this counter is activated can be selected by the setting of the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

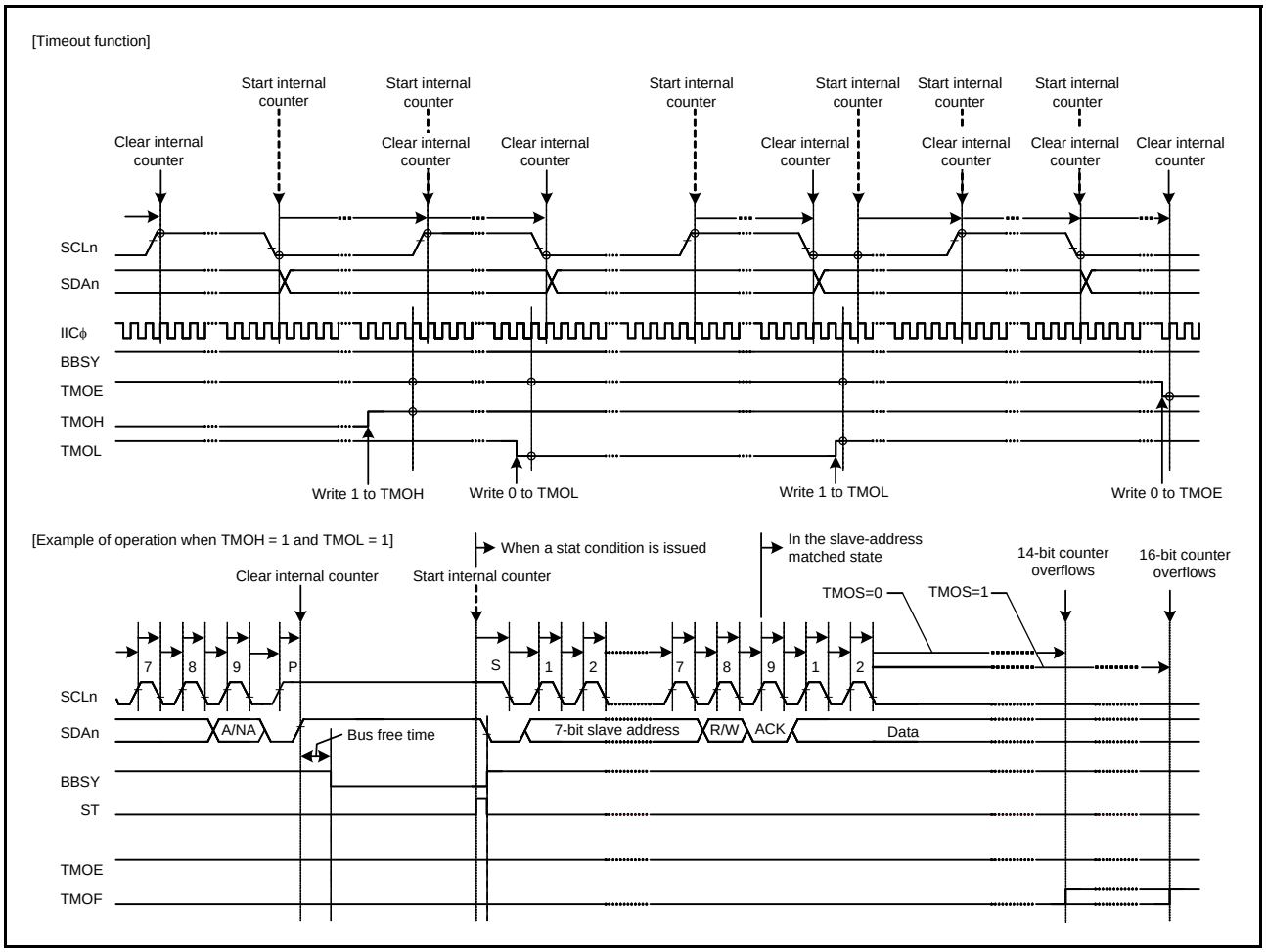


Figure 35.38 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

35.11.2 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 SCLn output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where 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 CLO bit in ICCR1 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 CKS[2:0] bits in ICMR1, and of the ICBRH and ICBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 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) signal 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 SDAI bit in ICCR1. After confirming release of the SDAn line by the slave device, complete communications by reissuing the stop condition.
Use this facility with the MALE bit (master arbitration-lost detection disabled) in ICFER cleared to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the SDAO bit in ICCR1does not match the state of the SDAn line, so take care on this point.
[Output conditions for using the CLO bit in ICCR1]

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCLn line low

Figure 35.39 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

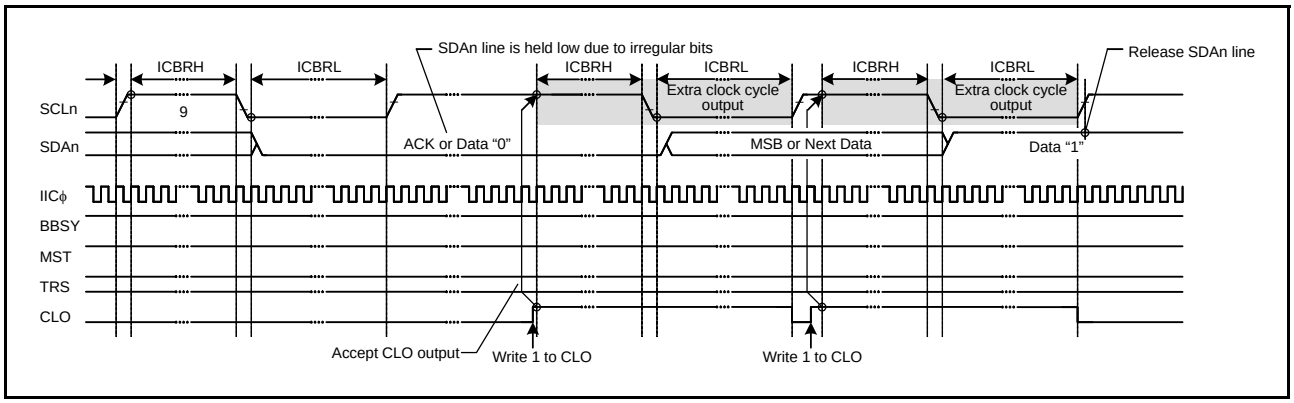


Figure 35.39 Extra SCL Clock Cycle Output Function (CLO Bit)

35.11.3 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 BBSY flag in ICCR2. 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 clear the IICRST bit in ICCR1 to 0.
Both types of reset are effective for release from bus-hung states since 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 (ICE and IICRST bits = 01b in ICCR1).
For a detailed description of the RIIC and internal resets, see section 35.14, Reset States.

35.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, ICBRH, and ICBRL. In addition, determine the values of the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 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. When the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL 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 (SARL0, SARL1, and SARL2), and set the corresponding FS bit (7-bit/10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

35.12.1 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 standard.
If the time measured with the MTU or TMR exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 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 SCLnn/SDAn pin outputs high impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: TLOW: 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 (ICTEI) or receive data full interrupt (ICRXI) 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 standard, 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 SMBCLK clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the RDRFS bit in ICMR3 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 SMBCLK 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 standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to ICDRT).

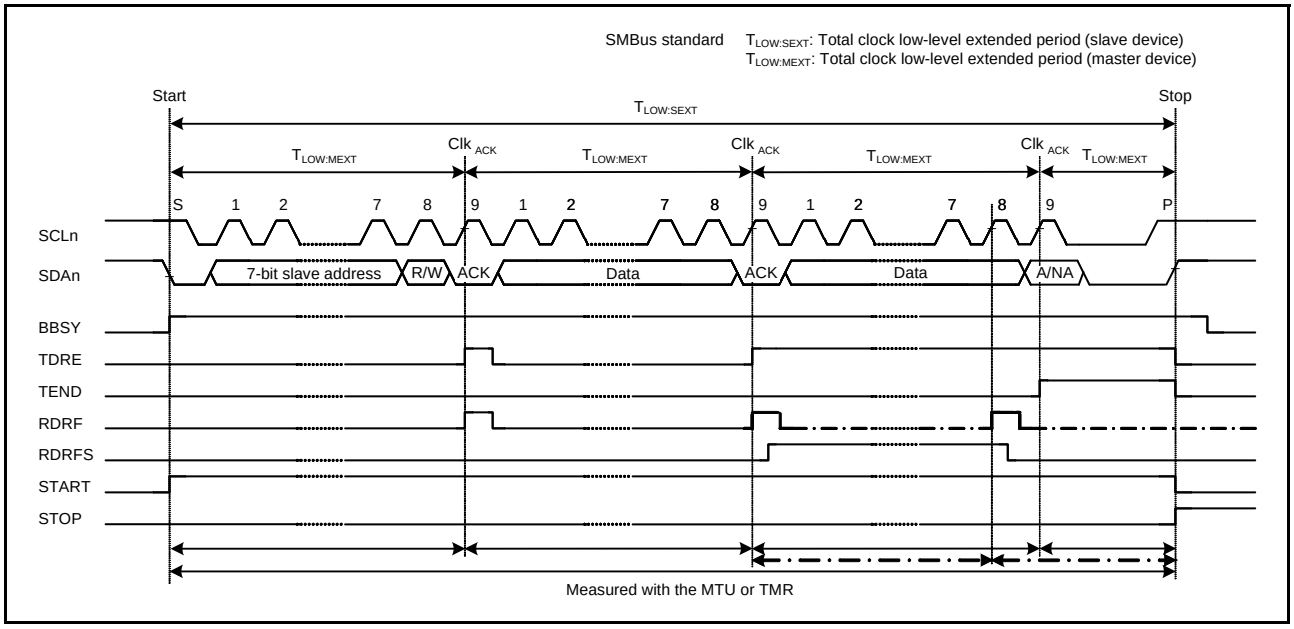


Figure 35.40 SMBus Timeout Measurement

35.12.2 Packet Error Code (PEC)

The RX63N/RX631 Group incorporates a CRC operation circuit. The CRC operation circuit 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 operation circuit, see section 39, CRC Calculator (CRC).

The PEC data in master transmit mode (master transmitter) can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC operation circuit.

The PEC data in master receive mode (master receiver) can be checked by writing all receive data to CRCDIR in the CRC operation circuit 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 RDRFS bit in ICMR3 to 1 before the rising edge of the eighth SMBCLK clock cycle during reception of the final byte, and hold the SCLn line low at the falling edge of the eighth clock cycle.

35.12.3 SMBus Host Notification Protocol/Notify ARP Master

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For a product of the RX63N/RX631 Group 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 SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address has been detected is the same as normal slave operation.

35.13 Interrupt Request

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 35.6 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of launching data transfer by the DTC or DMAC.

Table 35.6 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DTC Launching	DMACA Launching	Priority	Interrupt Condition
ICEEI	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
ICRXI	Receive Data Full	—	Possible	Possible		RDRF=1 • RIE=1
ICTXI	Transmit Data Empty	—	Possible	Possible		TDRE=1 • TIE=1
ICTEI	Transmit End	TEND	Not possible	Not possible	Low	TEND=1 • TEIE=1

Clear or mask the each flag during interrupt handling.

Notes on interrupt processing:

1. There is a latency (delay) 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 masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.
2. Since ICTXI is an edge-detected interrupted, it does not require clearing. Furthermore, the TDRE flag in ICSR2 (a condition for ICTXI) is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).
3. Since ICRXI is an edge-detected interrupted, it does not require clearing. Furthermore, the RDRF flag in ICSR2 (a condition for ICRXI) is automatically cleared to 0 when data are read from ICDRR.
4. When using the ICTEI interrupt, clear the TEND flag in ICSR2 in the ICTEI interrupt processing.
   Note that the TEND flag in ICSR2 is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

35.13.1 Buffer Operation for ICTXI and ICRXI Interrupts

For the ICTXI and ICRXI interrupts, as well as the ICU.IRn.IR flag having the value 1 being a condition for issuing the interrupts, the interrupt request is retained within and not output by the ICU (the capacity for internally retaining the interrupts is one request per source).

An interrupt request that was being retained within the ICU is output 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 interrupt enable bit within the given peripheral module.

35.14 Reset States

The RIIC has chip reset, RIIC reset, and internal reset functions. Table 35.7 lists the scope of each reset and reset conditions.

Table 35.7 Reset Conditions

		Chip Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	ICE, IICRST	At a reset	Retained	Retained	Operation (retained)	Operation (retained)
	SCLO, SDAO		At a reset	At a reset		
	Others			Retained		
ICCR2	BBSY	At a reset	At a reset	Operation	Operation	Operation
	ST			At a reset	At a reset	Operation (retained)
	Others					At a reset
ICMR1	BC[2:0]	At a reset	At a reset	At a reset	At a reset	Operation (retained)
	Others			Retained	Operation (retained)	
ICMR2		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICMR3		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICFER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICSER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICIER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICSR1		At a reset	At a reset	At a reset	Operation (retained)	At a reset
ICSR2	TDRE, TEND	At a reset	At a reset	At a reset	Operation (retained)	At a reset
	START				Operation	
	STOP				Operation (retained)	Operation
	Others					Operation (retained)
SARL0, 1, 2 SARU0, 1, 2		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICBRH, ICBRL		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRT		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRR		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRS		At a reset	At a reset	At a reset	Operation (retained)	Operation (retained)
Timeout function		At a reset	At a reset	Operation	Operation	Operation
Bus free time measurement		At a reset	At a reset	Operation	Operation	Operation

35.15 Usage Notes

35.15.1 Setting Module Stop Function

Module stop state can be entered or canceled 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 clearing module stop state.
For details of module stop control registers B and C, see section 11, Low Power Consumption.

35.15.2 Points to Note on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started (by setting the ICCR1.ICE bit to 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the ICU.IRn.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 ICU.IRn.IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Clear the relevant interrupt enable bits (ICIER1.TIE etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER1.TIE etc.) on the peripheral function side and confirm that its value is 0.
4. Clear the ICU.IRn.IR flag to 0.

36. CAN Module (CAN)

36.1 Overview

The RX63N/RX631 Group implements three channels of the CAN (Controller Area Network) module that complies with the ISO11898-1 Specifications. 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 36.1 lists the specifications of the CAN module, and Figure 36.1 shows a block diagram of the CAN module (i = 0 to 2).
Connect the CAN bus transceiver externally.

Table 36.1 Specifications of CAN Module

Item	Description
Protocol	• ISO11898-1 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: ISO11898-1 Specifications 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 (PCLK) or main clock
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.

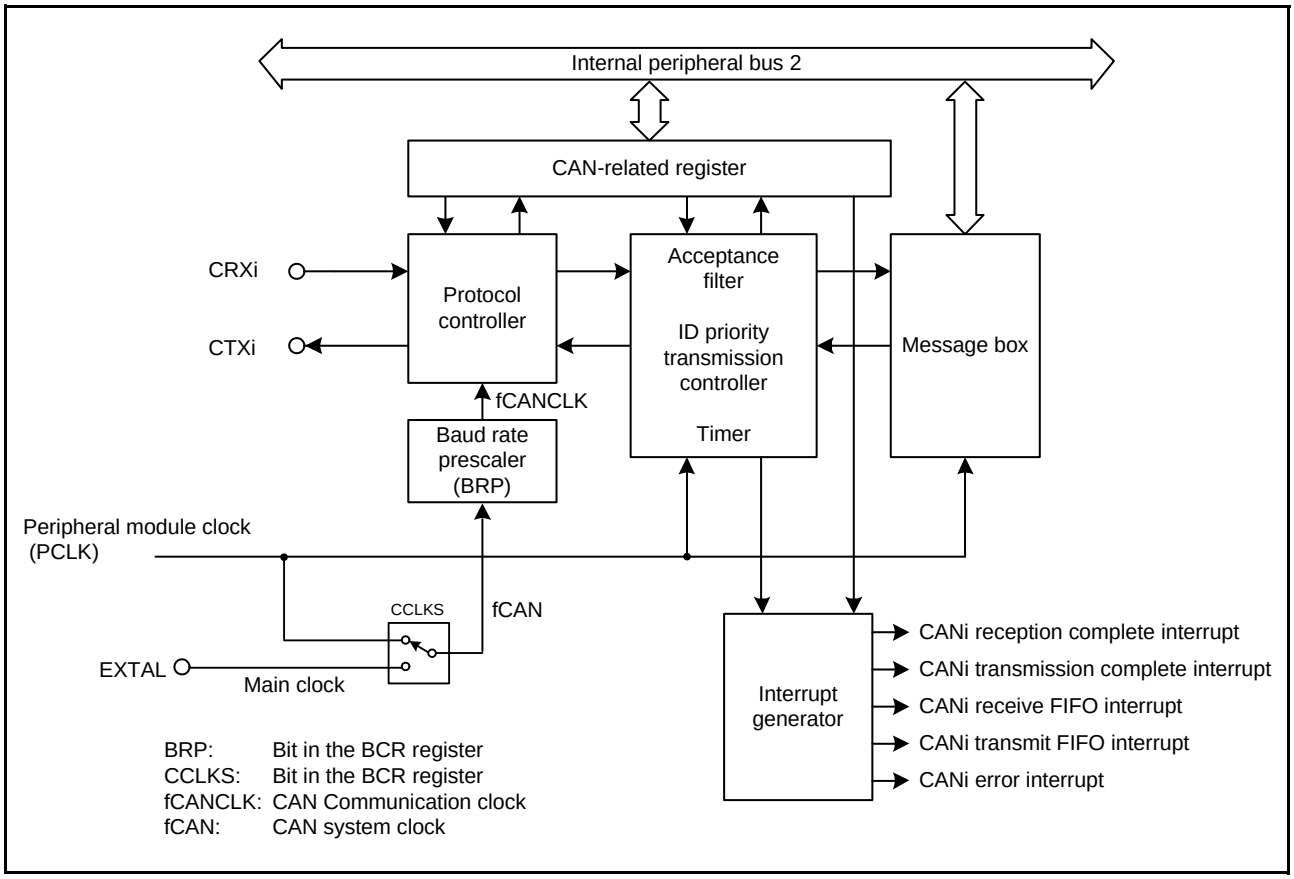


Figure 36.1 Block Diagram of CAN Module (i = 0 to 2)

- 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 36.2 lists the CAN module pins.
The CAN functions should be selected for the pins multiplexed with other signals. For details, see section 20, I/O Ports.

Table 36.2 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

36.2 Register Descriptions

36.2.1 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

Bit	Symbol	Bit Name	Description	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 (ISO11898-1 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 always 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 36.3 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 ISO11898-1 Specifications. 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 36.3, 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 36.3, 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 ISO11898-1 Specifications, 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 36.3 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 MIER1 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.

36.2.2 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: PLL clock 1: Main clock	R/W
b7	—	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: 1Tq 0 1: 2Tq 1 0: 3Tq 1 1: 4Tq	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	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 28	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

For bit timing setting, refer to section 36.4, 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 clock (PCLK) produced by the PLL frequency synthesizer is used as the CAN clock source (fCAN).
When the CCLKS bit is 1, the main clock signal externally input via 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)

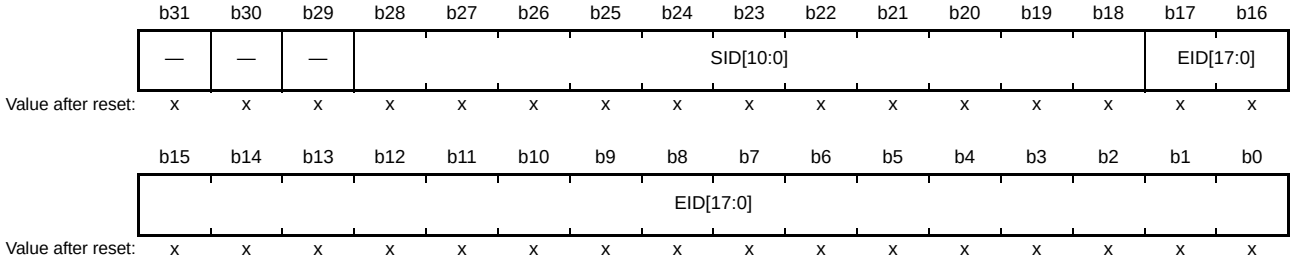
The BRP[9:0] bits are used to set the frequency of the CAN communication clock (fCANCLK). The fCANCLK cycle is 1 Tq. If the setting is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

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.

36.2.3 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



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 36.6, 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.

36.2.4 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.FIDCR0 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 36.6, 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 is 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.

36.2.5 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.

36.2.6 Mailbox Register j (MBj) (j = 0 to 31)

Table 36.4 lists the CANi mailbox memory mapping, and Table 36.5 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 36.4 for detailed register addresses.

Table 36.4 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

Table 36.5 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.MB63 0009 0200h to 0009 03FFh, CAN1.MB0 to CAN1.MB63 0009 1200h to 0009 13FFh, CAN2.MB0 to CAN2.MB63 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] bit is 0 1: Corresponding EID[17:0] bit is 1	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bit is 0 1: Corresponding SID[10:0] bit is 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

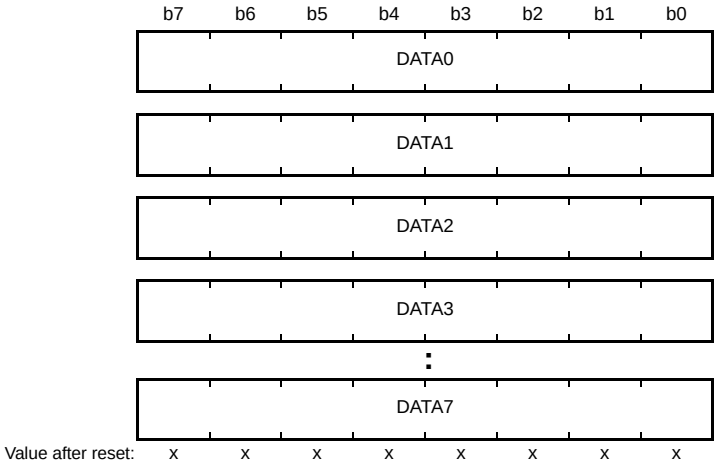
Note 1. If the mailbox has received a standard ID message, the EID bits in the mailbox are undefined.
Note 2. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode). When the IDFM[1:0] bits are not 10b, it should be written with 0 and read as 0.

	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 x: Represents any value.	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should always be 0.	R/W

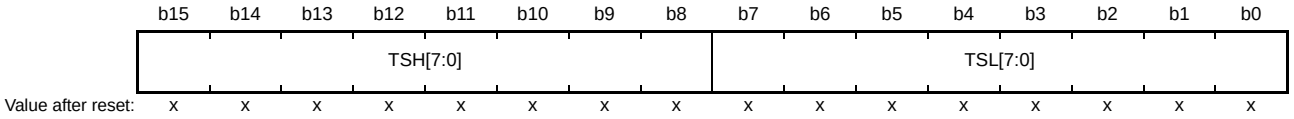
Note 1. If the mailbox has received a message whose data length set by the DLC[3:0] bits is less than 8 bytes, the values of DATA larger than the data length set by the DLC[3:0] bits in the mailbox are undefined.



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATA0 to DATA7	Data Bytes 0 to 7*1,*2	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 DATA0 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.



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.

36.2.7 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 MIER1 for the related FIFO only when the TFE bit in TFCR is 0 and the TFEST bit is 1, and the RFE bit in RFCR is 0 and the RFEST bit in RFCR is 1.

36.2.8 Message Control Register j (MCTLj) (j = 0 to 31)

Address(es): CAN0.MCTL0 to CAN0.MCTL31 0009 0820h to 0009 083F, CAN1.MCTL0 to CAN1.MCTL310009 1820h to 0009 183F, CAN2.MCTL0 to CAN2.MCTL31 0009 2820h to 0009 283F

- 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 bit 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.

NEWTDATA 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 bit. 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 bit is 1.

TRMACTIVE Bit (Transmission-in-Progress Status Flag)

The TRMACTIVE bit is set to 1 when the corresponding mailbox of the CAN module begins transmitting a message. The TRMACTIVE is set to 0 when the CAN module has lost CAN bus arbitration, a CAN bus error occurs, or data transmission is completed.

INVALDATA Bit (Reception-in-Progress Status Flag)

After the completion of a message reception, the INVALDATA bit is set to 1 while the received message is being updated into the corresponding mailbox. The INVALDATA bit is set to 0 immediately after the message has been stored. If the mailbox is read while the INVALDATA bit 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 (PCLK) 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 bits 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 bit is set to 1.

RECREQ Bit (Receive Mailbox Request)

The RECREQ bit selects receive modes listed in Table 36.10.
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 bits SENTDATA and TRMABT to 0 before changing to reception.

TRMREQ Bit (Transmit Mailbox Request)

The TRMREQ bit selects transmit modes listed in Table 36.10.
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 bits NEWDATA and MSGLOST to 0 before changing to transmission.

36.2.9 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-b1	RFUST[2:0]	Receive FIFO Unread Message Number Status	b3 b2 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 bit = 1). Write 0 to the RFE bit simultaneously with setting the RFMLF bit.
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] Bits (Receive FIFO Unread Message Number Status)

The RFUST[2:0] bits indicate the number of unread messages in the receive FIFO.
The value of the RFUST[2:0] bits is initialized to 000b when the RFE bit is set to 0.

RFMLF Flag (Receive FIFO Message Lost Flag)

The RFMLF bit 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 bit is set to 0 by writing 0 by a program (writing 1 has no effect). In both overwrite and overrun modes, the RFMLF bit 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 (PCLK) 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 bit is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. The RFFST bit is 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. The RFFST bit is set to 0 when the RFE bit is 0.

RFWST Flag (Receive FIFO Buffer Warning Status Flag)

The RFWST bit is set to 1 (receive FIFO is buffer warning) when the number of unread messages in the receive FIFO is 3. The RFWST bit 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 bit is set to 0 when the RFE bit is 0.

RFEST Flag (Receive FIFO Empty Status Flag)

The RFEST bit is set to 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. The RFEST bit is set to 1 when the RFE bit is set to 0. The RFEST bit is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more.

Figure 36.2 shows the receive FIFO mailbox operation.

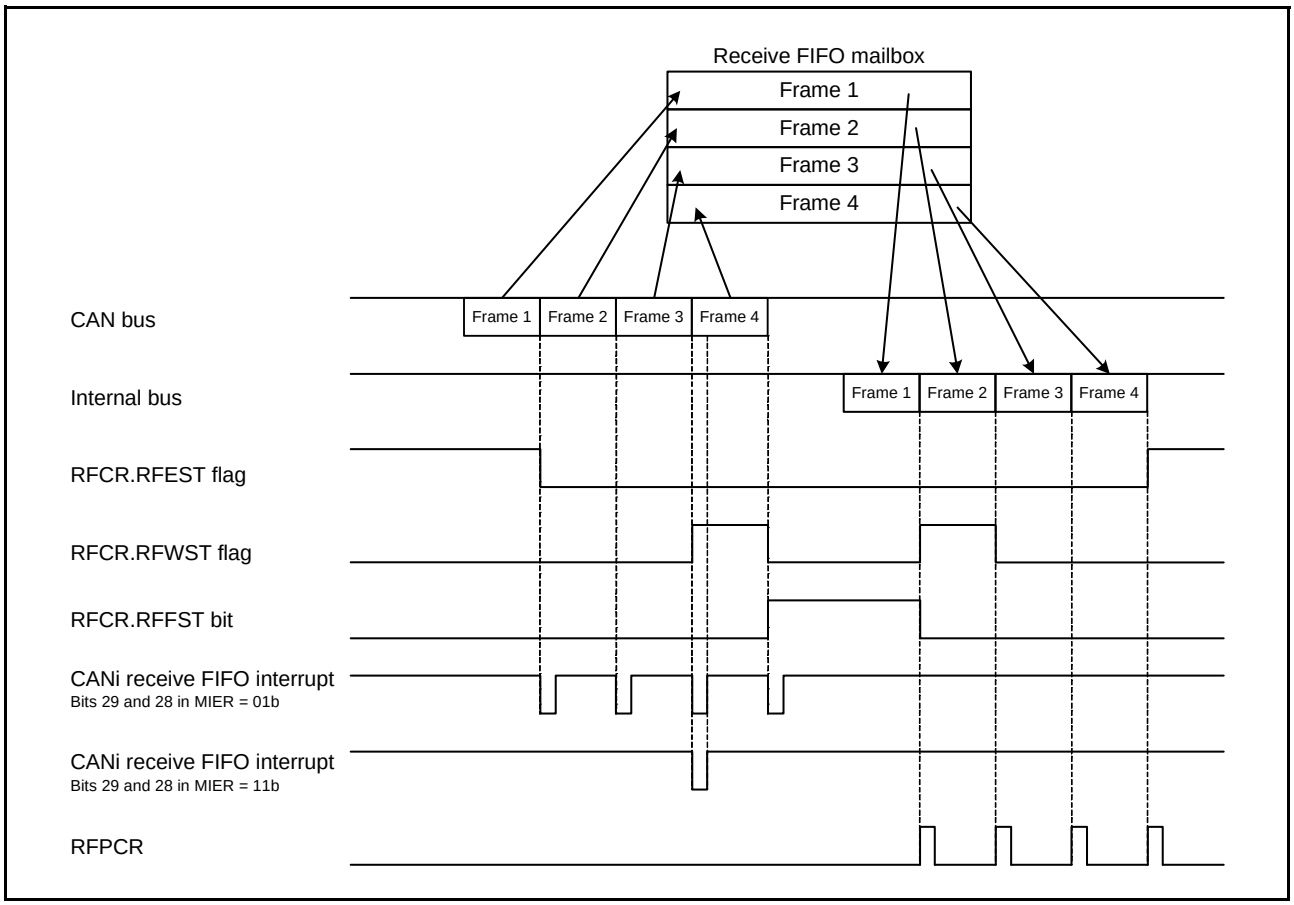


Figure 36.2 Receive FIFO Mailbox Operation (Bits 29 and 28 in MIER = 01b or 11b)

36.2.10 Receive FIFO Pointer Control Register (RFPCR)

Address(es): CAN0.RFPCR 0009 0849h, CAN1.RFPCR 009 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 bit is 1 (receive FIFO is full) in overwrite mode. When the RFMLF bit is 1 in this condition, the CPU-side pointer cannot be incremented by writing to RFPCR by a program.

36.2.11 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	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	0: Transmit FIFO is not full 1: Transmit FIFO is full (4 unsent messages)	R
b7	TFEST	Transmit FIFO Full Status	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 bit = 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 bit has been set to 1. After setting the TFE bit to 1, write transmit data into MB23.
Do not set the TFE bit to 1 in normal mailbox mode (MBM bit in CTLR = 0).

TFUST[2:0] Bits (Transmit FIFO Unsent Message Number Status)

The TFUST[2:0] bits indicate the number of unsent messages in the transmit FIFO.
The TFUST[2:0] bits are set to 000b after TFE bit is cleared to 0 and transmission is aborted or completed.

TFFST Bit (Transmit FIFO Full Status)

The TFFST bit is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. The TFFST bit 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 bit is set to 0 when transmission from the transmit FIFO has been aborted.

TFEST Bit (Transmit FIFO Empty Status)

The TFEST bit is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0. The TFEST bit is set to 1 when transmission from the transmit FIFO has been aborted. The TFEST bit is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 36.3 shows the transmit FIFO mailbox operation.

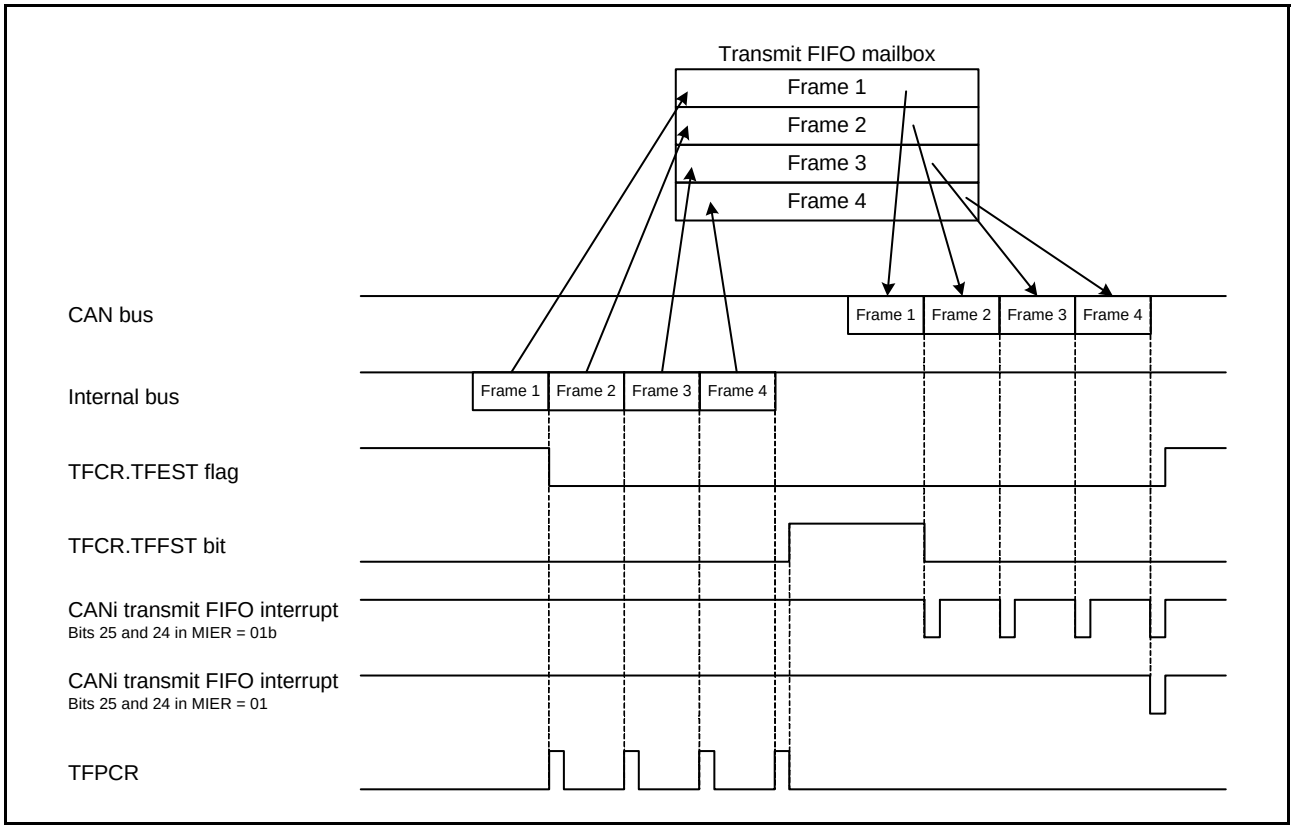


Figure 36.3 Transmit FIFO Mailbox Operation (Bits 25 and 24 in MIER = 01b or 11b)

36.2.12 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).

36.2.13 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 bit = 1 1: Mailbox(es) with NEWDATA bit = 1	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA bit = 1 1: Mailbox(es) with SENTDATA bit = 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 bit = 1 1: Mailbox(es) with MSGLOST bit = 1	R
b5	FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF bit = 0 1: RFMLF bit = 1	R
b6	TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT bit = 1 1: Mailbox(es) with TRMABT bit = 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 bit is set to 1 when at least one NEWDATA bit in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The NDST bit is set to 0 when all NEWDATA bits are 0.

SDST Flag (SENTDATA Status Flag)

The SDST bit is set to 1 when at least one SENTDATA bit in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The SDST bit is set to 0 when all SENTDATA bits are 0.

RFST Flag (Receive FIFO Status Flag)

The RFST bit is set to 1 when the receive FIFO is not empty. The RFST bit is set to 0 when the receive FIFO is empty or normal mailbox mode is selected.

TFST Flag (Transmit FIFO Status Flag)

The TFST bit is set to 1 when the transmit FIFO is not full. The TFST bit 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 bit is set to 1 when at least one MSGLOST bit in MCTLj (i = 0 to 2, j = 0 to 31) is 1 regardless of the value of MIER. The NMLST bit is set to 0 when all MSGLOST bits are 0.

FMLST Flag (FIFO Mailbox Message Lost Status Flag)

The FMLST bit is set to 1 when the RFMLF bit in RFCR is 1 regardless of the value of MIER. The FMLST bit is set to 0 when the RFMLF bit is 0.

TABST Flag (Transmission Abort Status Flag)

The TABST bit is set to 1 when at least one TRMABT bit in MCTLj (i = 0 to 2, j = 0 to 31) is 1 regardless of the value of MIER. The TABST bit is set to 0 when all TRMABT bits are 0.

EST Flag (Error Status Flag)

The EST bit is set to 1 when at least one error is detected by EIFR regardless of the value of EIER. The EST bit is set to 0 when no error is detected by EIFR.

RSTST Flag (CAN Reset Status Flag)

The RSTST bit is set to 1 when the CAN module is in CAN reset mode. The RSTST bit 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 bit remains 1.

HLTST Flag (CAN Halt Status Flag)

The HLTST bit is set to 1 when the CAN module is in CAN halt mode. The HLTST bit 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 bit remains 1.

SLPST Flag (CAN Sleep Status Flag)

The SLPST bit is set to 1 when the CAN module is in CAN sleep mode. The SLPST bit is set to 0 when the CAN module is not in CAN sleep mode.

EPST Flag (Error-Passive Status Flag)

The EPST bit 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 bit is set to 0 when the CAN module is not in the error-passive state.

BOST Flag (Bus-Off Status Flag)

The BOST bit 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 bit is set to 0 when the CAN module is not in the bus-off state.

TRMST Flag (Transmit Status Flag) (transmitter)

The TRMST bit is set to 1 when the CAN module performs as a transmitter node or is in the bus-off state. The TRMST bit 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 bit is set to 1 when the CAN module performs as a receiver node. The RECST bit is set to 0 when the CAN module performs as a transmitter node or is in the bus-idle state.

36.2.14 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 bit in MCTLj (j = 0 to 31) for the normal mailbox and the RFEST bit 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 bit 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 bit in MCTLj for the normal mailbox and the RFMLF bit 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 36.2.16, Channel Search Support Register (CSSR).

36.2.15 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	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	0: Search result found 1: No search result	R

MBNST[4:0] Bits (Search Result Mailbox Number Status)

The MBNST[4:0] bits 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 bit for the output mailbox is set to 0
- When the NEWDATA, SENTDATA or MSGLOST bit 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 36.6 lists the behavior of the MBNST[4:0] bits in FIFO mailbox mode.

In channel search mode, the MBNST[4:0] bits output the corresponding channel number. After MSSR is read by a program, the next target channel number is output.

SEST Bit (Search Result Status)

The SEST bit 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 bit is set to 1 when no SENTDATA bit for mailboxes is 1. The SEST bit is set to 0 when at least one SENTDATA bit is 1. When the SEST bit is 1, the value of the MBNST[4:0] bits is undefined.

Table 36.6 Behavior of MBNST[4:0] Bits 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 bit 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 bit 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.

36.2.16 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] bits 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 36.4 shows the write and read of CSSR and MSSR.

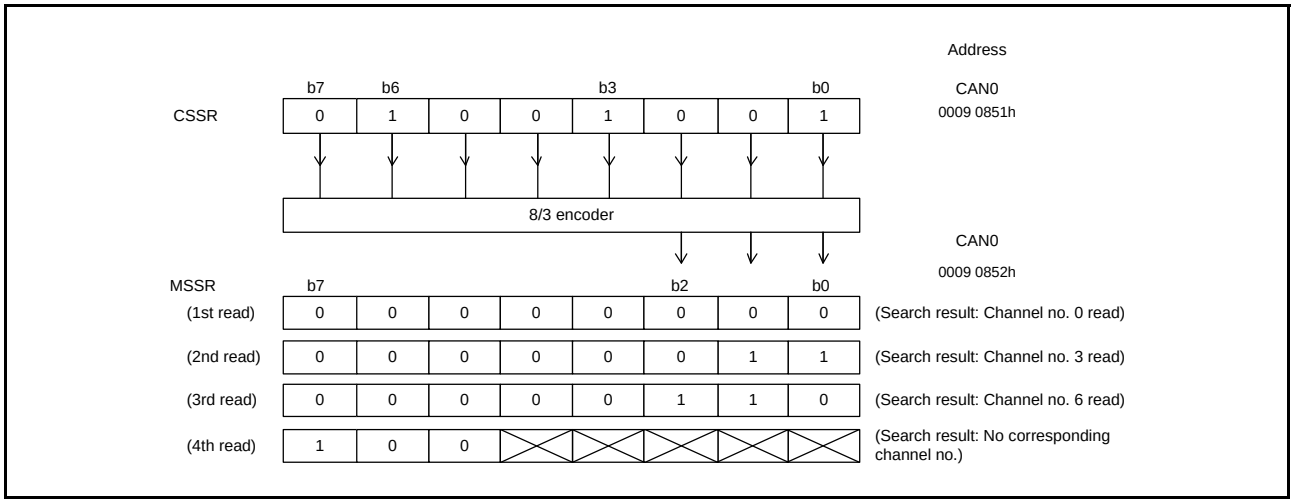


Figure 36.4 Write and Read of CSSR and MSSR

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.

36.2.17 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 36.5 shows the write and read of AFSR.

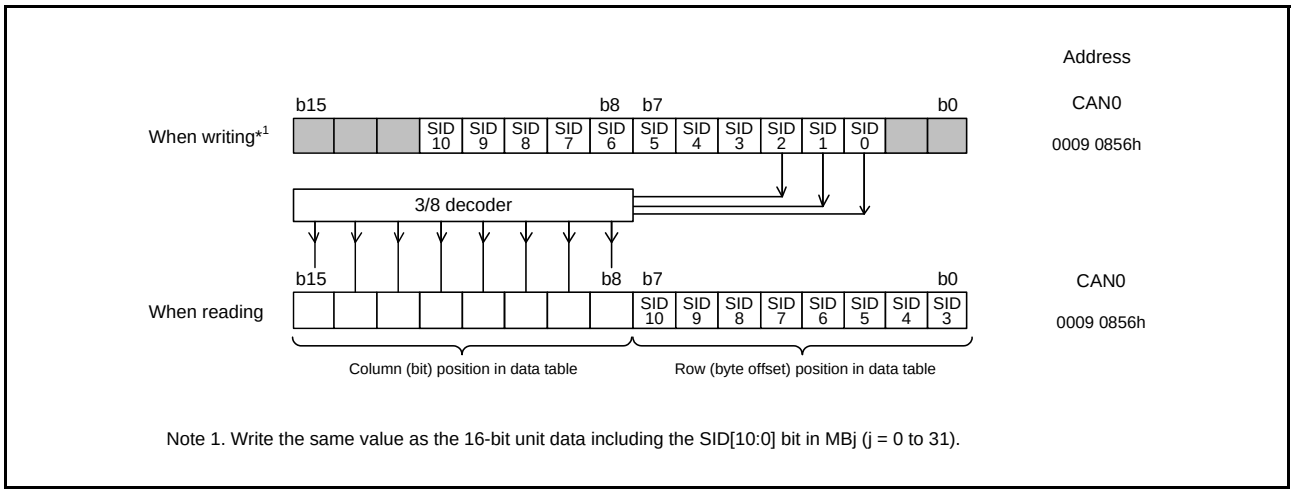


Figure 36.5 Write and Read of AFSR

36.2.18 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 bit in EIFR is set to 1. When the BEIE bit is 1, an error interrupt request is generated if the BEIF bit 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 bit in EIFR is set to 1. When the EWIE bit is 1, an error interrupt request is generated if the EWIF bit 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 bit in EIFR is set to 1. When the EPIE bit is 1, an error interrupt request is generated if the EPIF bit 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 bit in EIFR is set to 1. When the BOEIE bit is 1, an error interrupt request is generated if the BOEIF bit 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 bit in EIFR is set to 1. When the BORIE bit is set to 1, an error interrupt request is generated if the BORIF bit 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 bit in EIFR is set to 1. When the ORIE bit is 1, an error interrupt request is generated if the ORIF bit 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 bit in EIFR is set to 1. When the OLIE bit is 1, an error interrupt request is generated if the OLIF bit 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 bit in EIFR is set to 1. When the BLIE bit is 1, an error interrupt request is generated if the BLIF bit is set to 1.

36.2.19 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 bit is set to 1 when a bus error is detected.

EWIF Flag (Error-Warning Detect Flag)

The EWIF bit is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95. The EWIF bit is set to 1 only when the REC or TEC initially exceeds 95. Thus, if 0 is written to the EWIF bit by a program while the REC or TEC remains greater than 95, the EWIF bit 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 bit 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 bit 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 bit 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 bit is set to 1 when the CAN error state becomes bus-off (the TEC (transmit error counter) value exceeds 255). The BOEIF bit 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 bit 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 bit 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

ORIF Flag (Receive Overrun Detect Flag)

The ORIF bit is set to 1 when a receive overrun occurs. This bit 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 bit is not set to 1.
In normal mailbox mode, if an overrun occurs in any of mailboxes [0] to [31] in overrun mode, this bit 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 bit is set to 1.

OLIF Flag (Overload Frame Transmission Detect Flag)

The OLIF bit 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 bit 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 bit is set to 1, 32 consecutive dominant bits are detected again under either of the following conditions:

- After this bit is set to 0 from 1, recessive bits are detected
- After this bit is set to 0 from 1, the CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again.

Table 36.7 lists the behavior of bits BOEIF and BORIF according to the CTLR.BOM[1:0] bit setting.

Table 36.7 Behavior of BOEIF and BORIF Flags according to CTLR.BOM[1:0] Bit Setting

BOM[1:0] Bits	BOEIF Bit	BORIF Bit
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).

36.2.20 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 CiRECR 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 CAN Specifications (ISO11898-1) about the increment/decrement conditions of the receive error counter.

The value of RECR in the bus-off state is undefined.

36.2.21 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 CiTECR 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 CAN Specifications (ISO11898-1) about the increment/decrement conditions of the transmit error counter.
The value of TECR in the bus-off state is undefined.

36.2.22 Error Code Store Register (ECSR)

Address(es): CAN0.ECSR 0000 0850h, CAN1.ECSR 0000 1850h, CAN2.ECSR 0000 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 bit values.
Note 2. To write 0 to bits SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF, do not use the logic operation instruction (AND). Use the transfer (MOV) instruction to ensure that only the specified bit is set to 0 and the other bits 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 bits are set to 1.

ECSR can be used to monitor whether an error has occurred on the CAN bus.
Refer to the CAN Specifications (ISO11898-1) to check the generation conditions of each error.
To set each bit except for the EDPM bit to 0, write 0 by a program. If the timing at which each bit is set to 1 and the timing at which 0 is written by a program are the same, the relevant bit is set to 1.

SEF Flag (Stuff Error Flag)

The SEF bit is set to 1 when a stuff error is detected.

FEF Flag (Form Error Flag)

The FEF bit is set to 1 when a form error is detected.

AEF Flag (ACK Error Flag)

The AEF bit is set to 1 when an ACK error is detected.

CEF Flag (CRC Error Flag)

The CEF bit is set to 1 when a CRC error is detected.

BE1F Flag (Bit Error (recessive) Flag)

The BE1F bit is set to 1 when a recessive bit error is detected.

BE0F Flag (Bit Error (dominant) Flag)

The BE0F bit is set to 1 when a dominant bit error is detected.

ADEF Flag (ACK Delimiter Error Flag)

The ADEF bit 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.

36.2.23 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.

36.2.24 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 CAN Specifications (ISO11898-1) 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 36.6 shows the connection when listen-only mode is selected (i = 0 to 2).

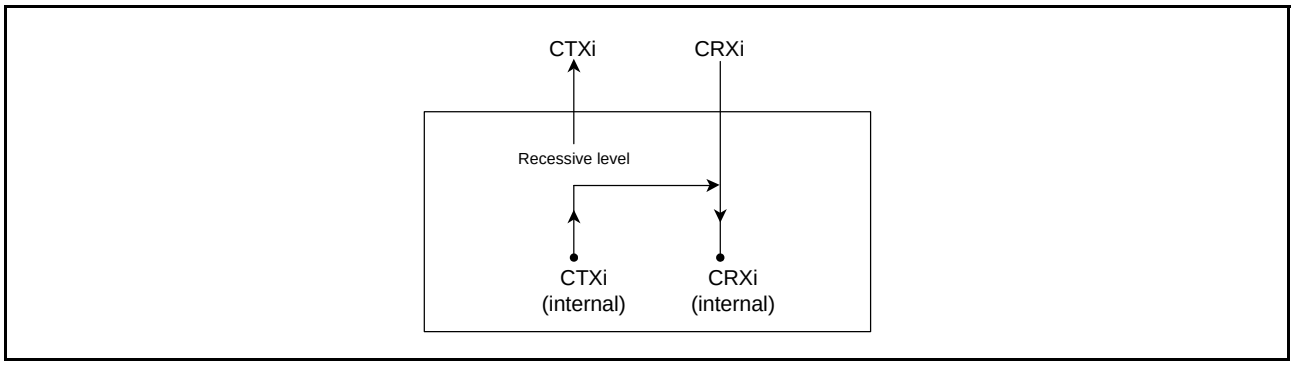


Figure 36.6 Connection when Listen-Only Mode is Selected (i = 0 to 2)

(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 36.7 shows the connection when self-test mode 0 is selected (i = 0 to 2).

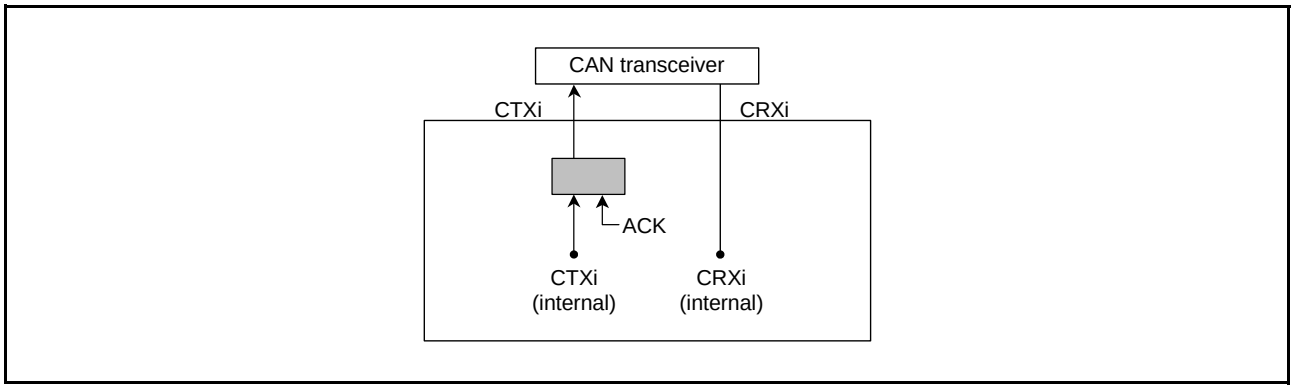


Figure 36.7 Connection when Self-Test Mode 0 is Selected (i = 0 to 2)

(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 36.8 shows the connection when self-test mode 1 is selected (i = 0 to 2).

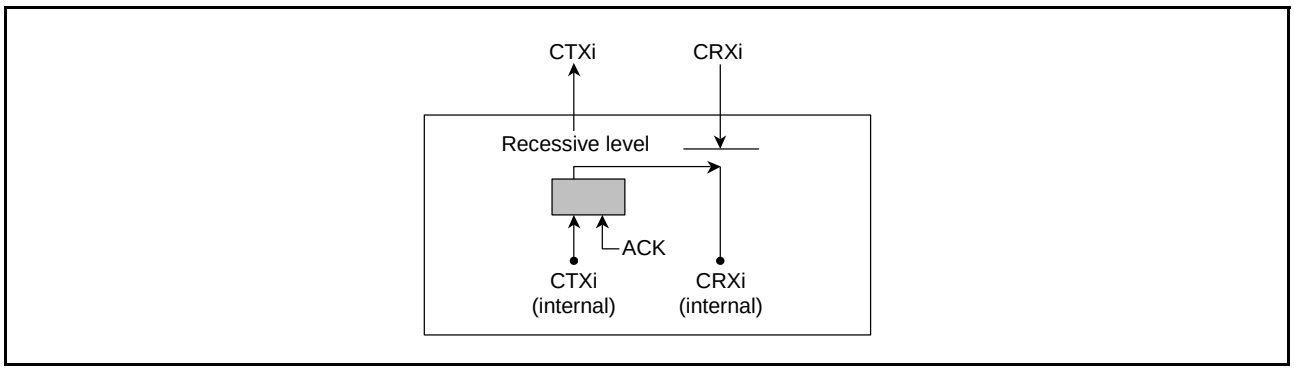


Figure 36.8 Connection when Self-Test Mode 1 is Selected (i = 0 to 2)

36.3 Operating Mode

The CAN module has the following four operating modes.

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode

Figure 36.9 shows the transition between CAN operating modes.

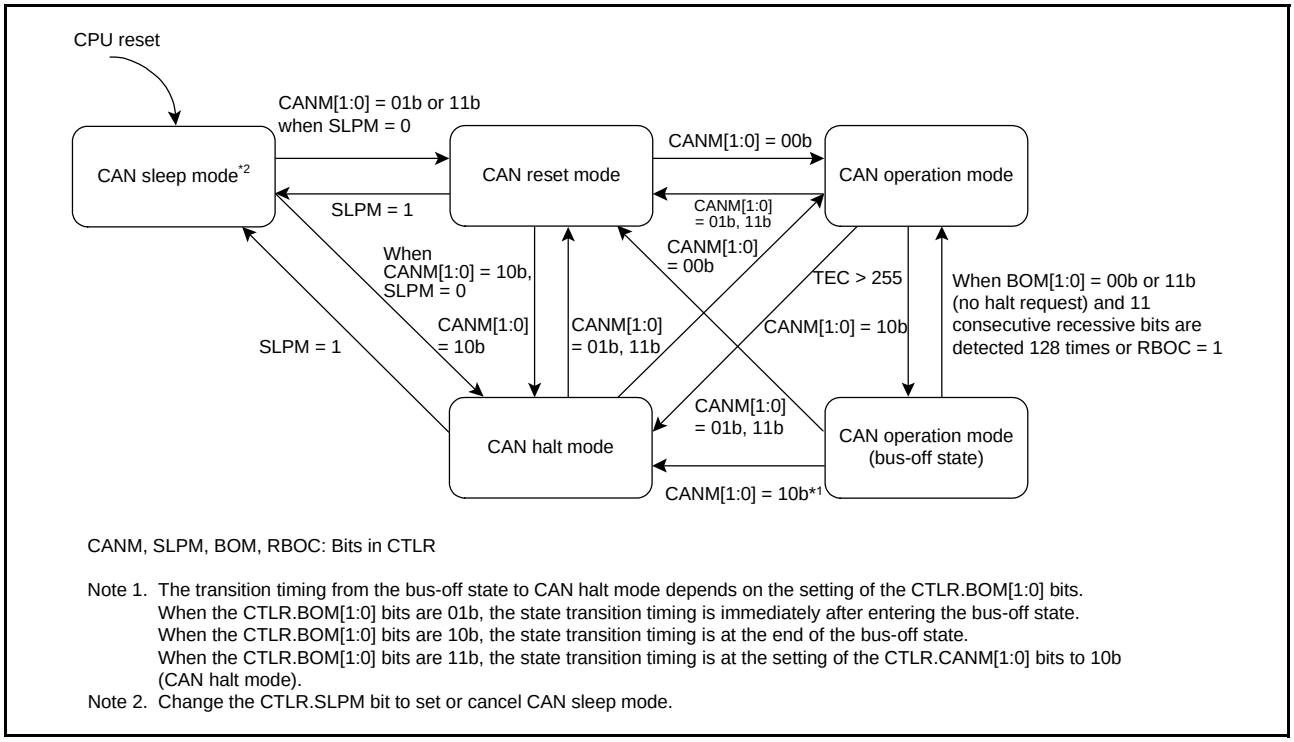


Figure 36.9 Transition between CAN Operating Modes

36.3.1 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 bit is set to 1. Do not change the CTLR.CANM[1:0] bits until the RSTST bit 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 bits)
- 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 bits)
- MIER
- EIER
- BCR
- CSSR
- ECSR (only the EDPM bit)
- MBj
- MKR0 to MKR7
- FIDCR0 and FIDCR1
- MKIVLR
- AFSR
- RFPCR
- TFPCR

36.3.2 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 bit is set to 1. Do not change the CTLR.CANM[1:0] bits until the HLTST bit is set to 1.
See Table 36.8 for the state transition conditions when transmitting or receiving.
All registers except for bits RSTST, HLTST, and SLPST 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 36.8 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,*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.

BOM[1:0] bits: Bits in CTLR

Note 1. If several messages are requested to be transmitted, mode transition occurs after the completion of the first 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 BLIF bit in EIFR.

Note 3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN module transits to 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 CAN mode.

36.3.3 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 bit in STR is set to 1. Do not change the value of the SLPM bit until the SLPST bit 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.

36.3.4 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 bits RSTST and HLTST in STR are set to 0. Do not change the value of the CANM[1:0] bits until bits RSTST and HLTST 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 36.10 shows the sub-modes of CAN operation mode.

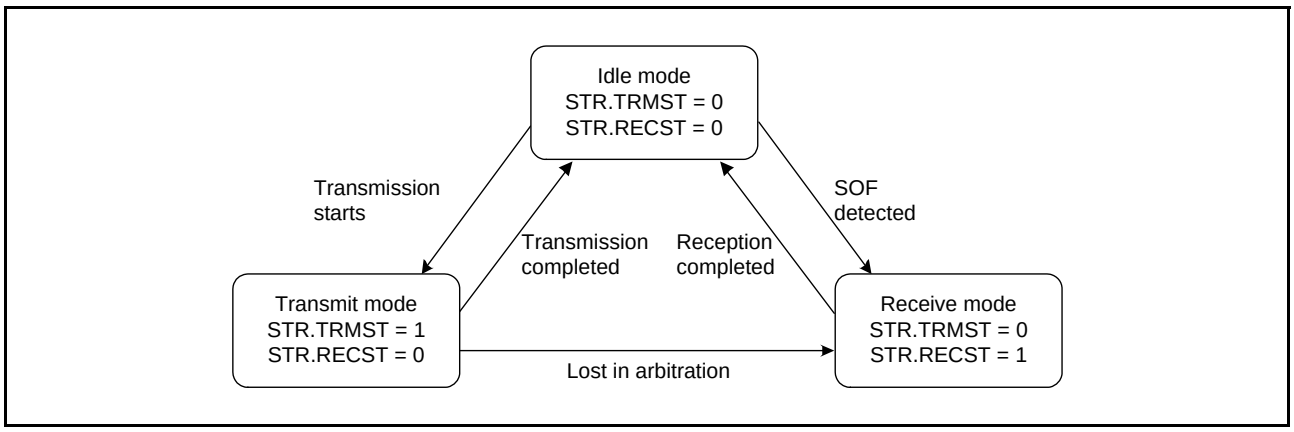


Figure 36.10 Sub-Modes of CAN Operation Mode

36.3.5 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 bit 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 bit 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 bit 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 bit 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.

36.4 CAN Communication Speed Setting

The following description explains about CAN communication speed setting.

36.4.1 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 36.11 shows a block diagram of the CAN clock generator.

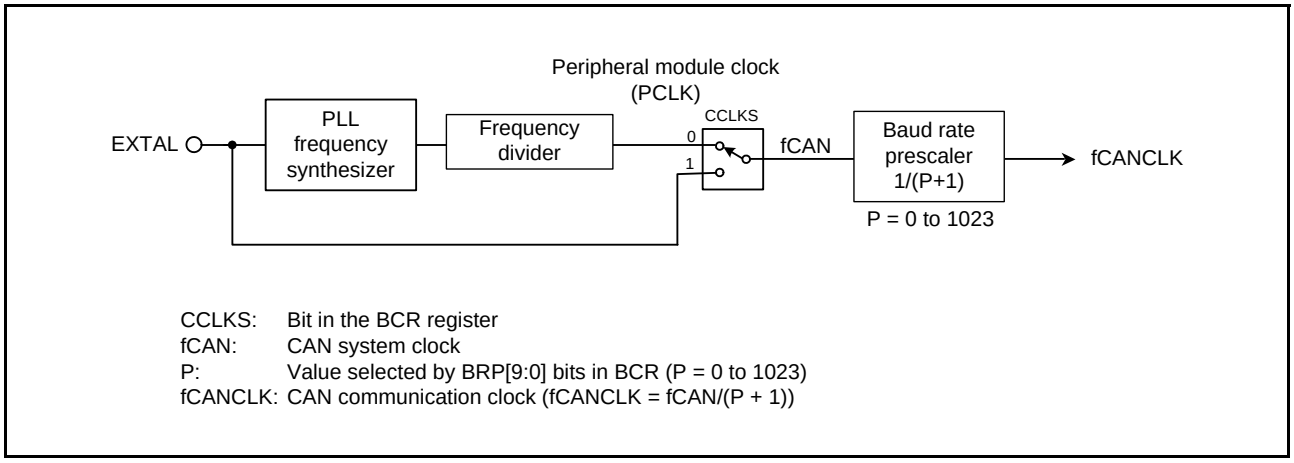


Figure 36.11 Block Diagram of CAN Clock Generator

36.4.2 Bit Timing Setting

The bit time consists of the following three segments.
Figure 36.12 shows the bit timing.

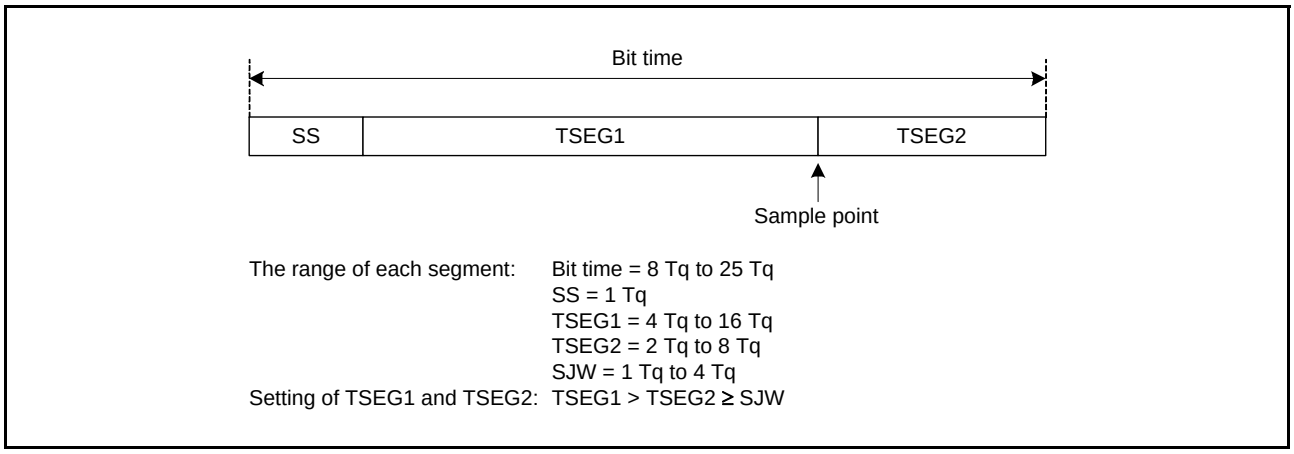


Figure 36.12 Bit Timing

36.4.3 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.

$$\text{Bit rate [bps]} = \frac{\text{fCAN}}{\text{Baud rate prescaler division value*1 x number of Tq of 1 bit time}} = \frac{\text{fCANCLK}}{\text{Number of Tq of 1 bit time}}$$

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 36.9 lists bit rate examples.

Table 36.9 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

36.5 Mailbox and Mask Register Structure

Figure 36.13 shows the structure of MBj.
There are 32 mailboxes with the same structure.

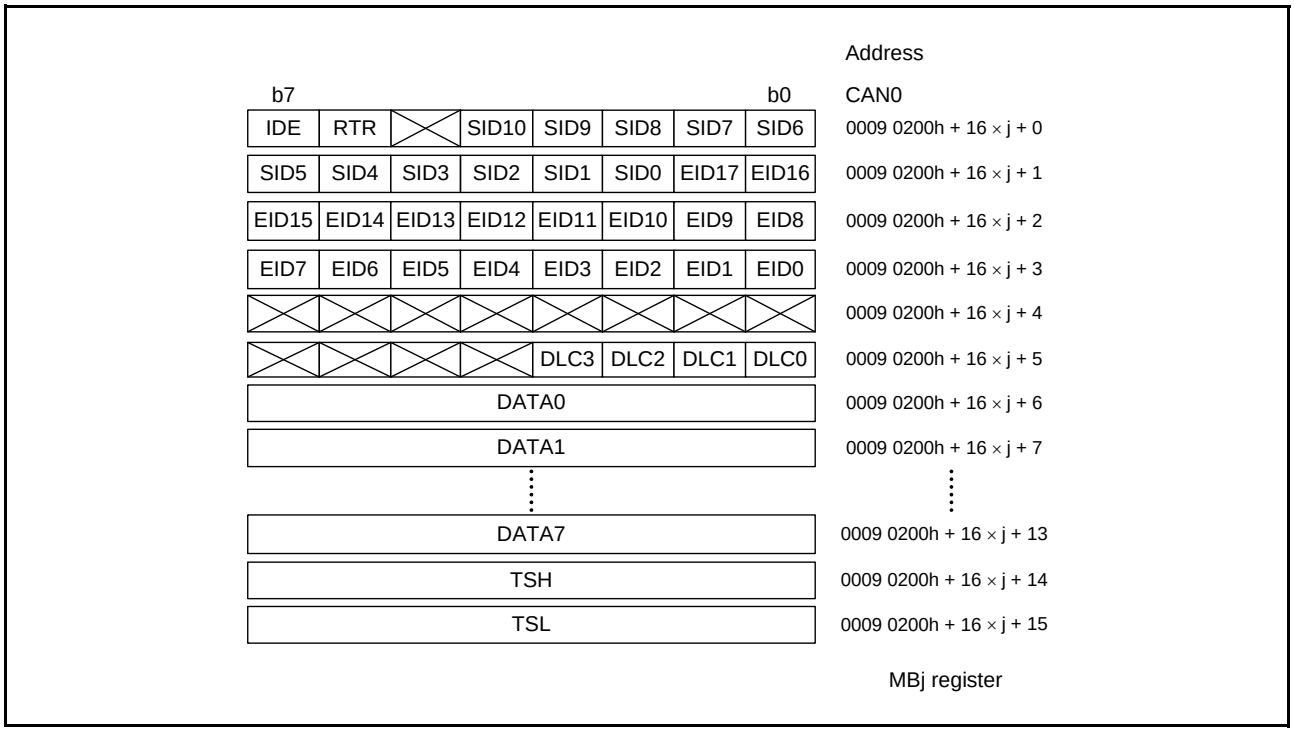


Figure 36.13 Structure of MBj (j = 0 to 31)

Figure 36.14 shows the structure of MKRk.
There are eight mask registers with the same structure.

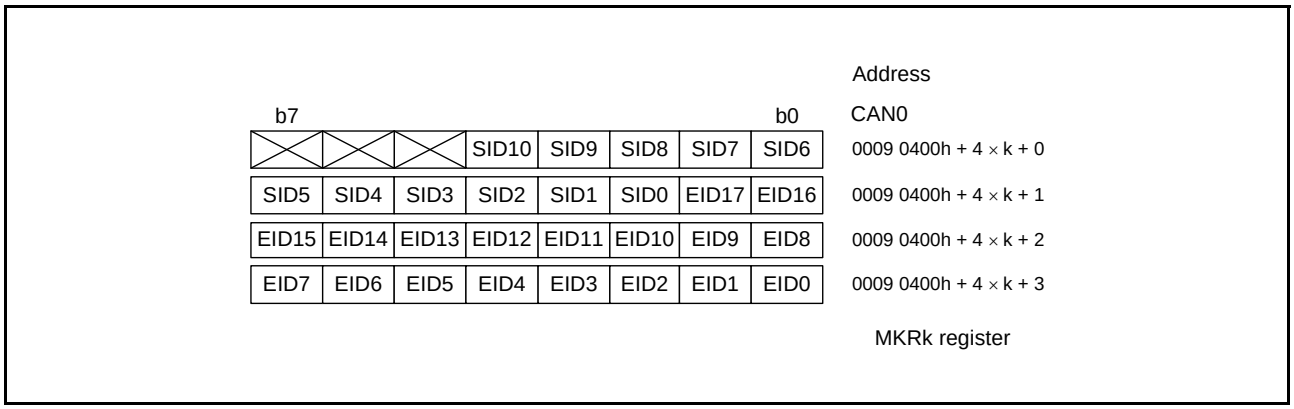


Figure 36.14 Structure of MKRk (k = 0 to 7)

Figure 36.15 shows the structure of FIDCR0 and FIDCR1.
There are two FIFO received ID compare registers with the same structure.

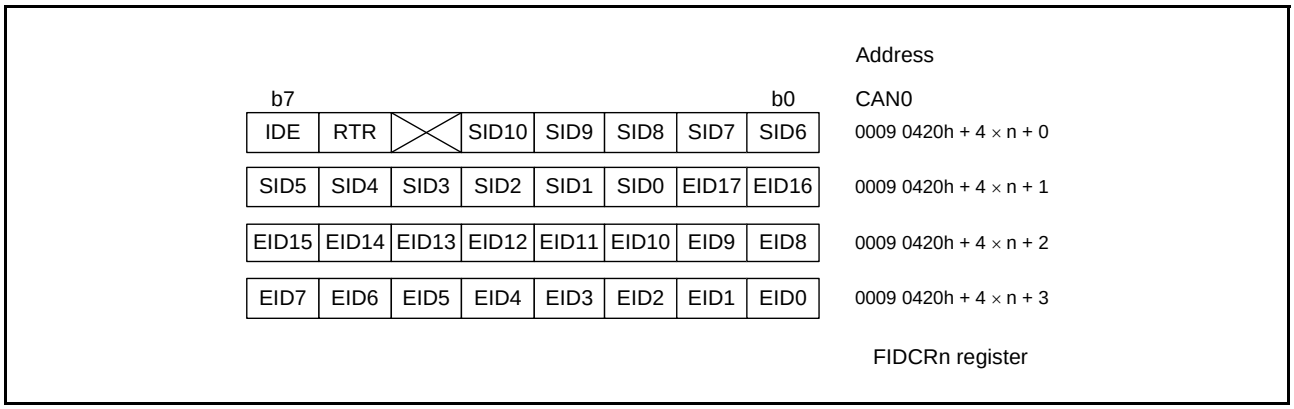


Figure 36.15 Structure of FIDCRn (n = 0, 1)

36.6 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 36.16 shows the correspondence between mask registers and mailboxes. Figure 36.17 shows acceptance filtering.

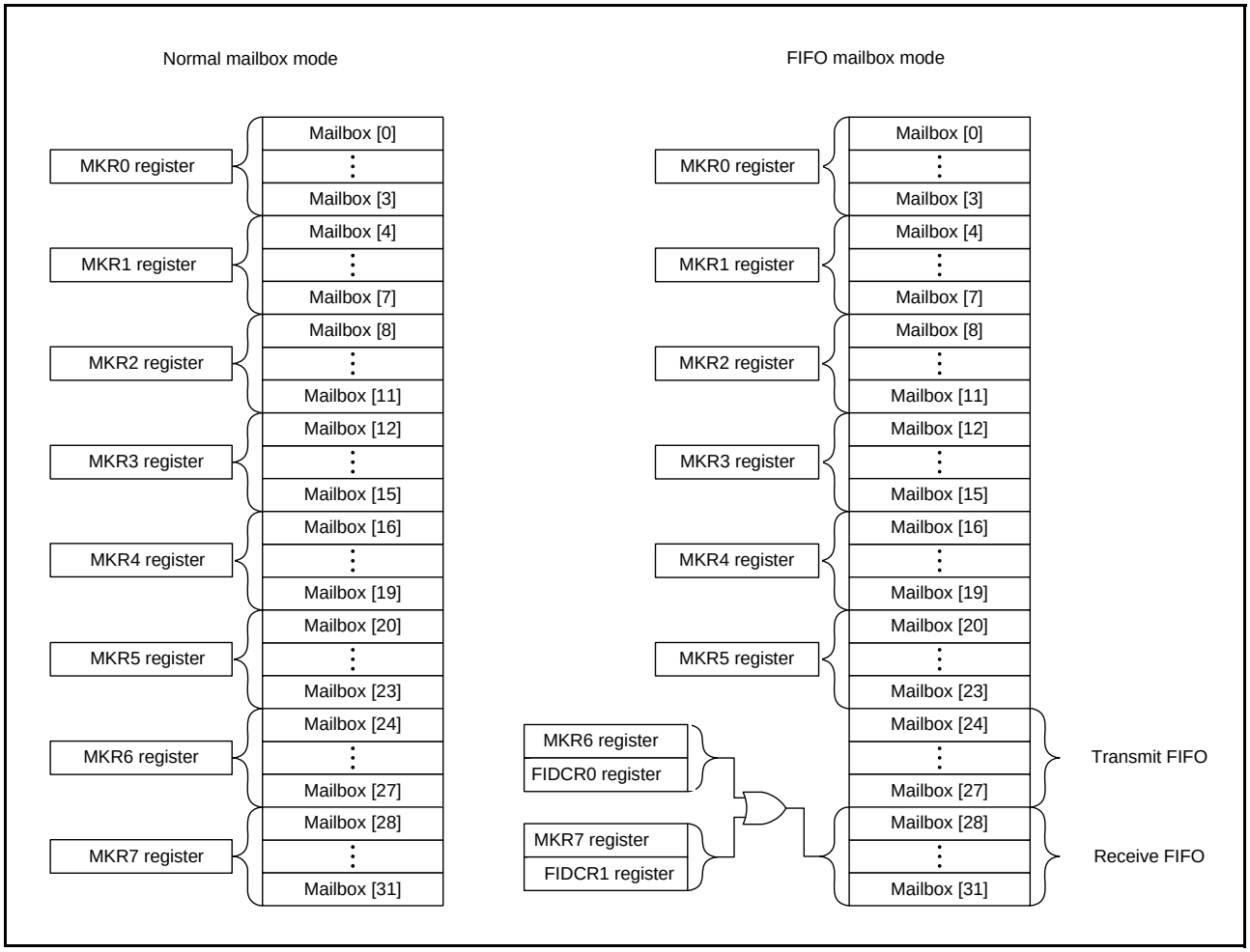
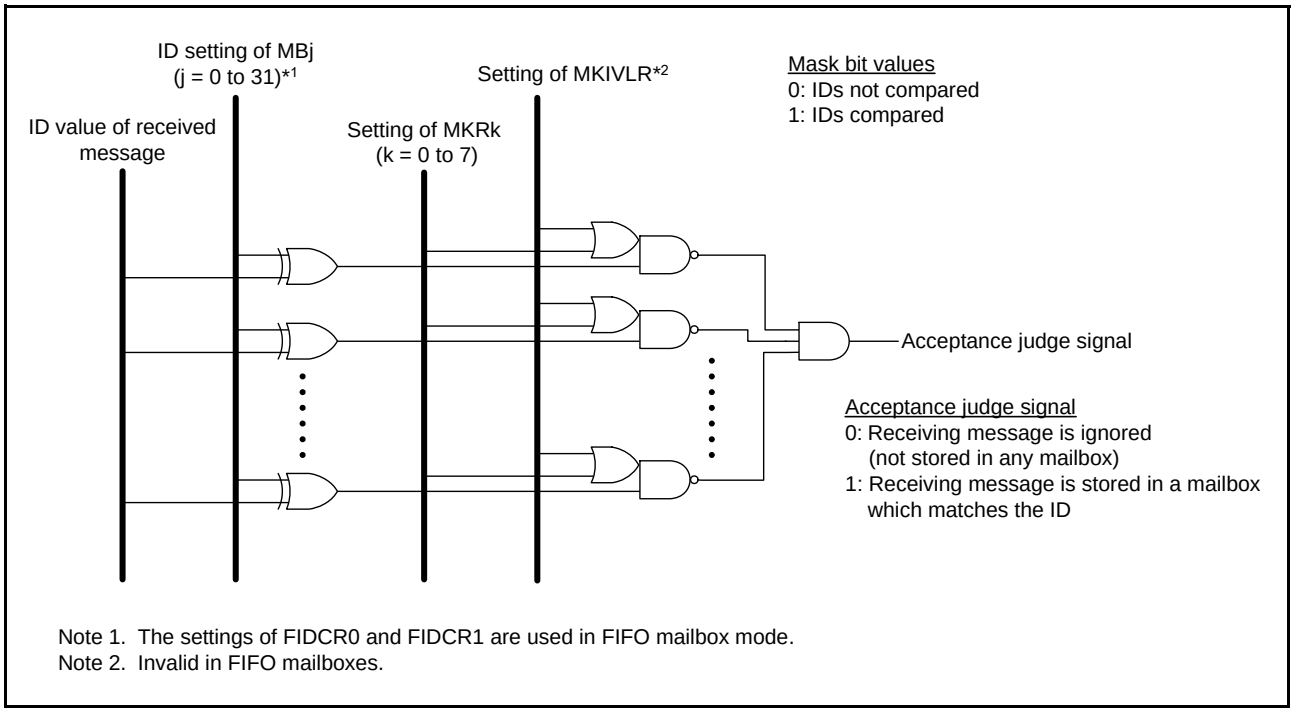


Figure 36.16 Correspondence between Mask Registers and Mailboxes



Note 1. The settings of FIDCR0 and FIDCR1 are used in FIFO mailbox mode.
Note 2. Invalid in FIFO mailboxes.

Figure 36.17 Acceptance Filtering

36.7 Reception and Transmission

Table 36.10 lists how to make the CAN communication mode settings.

Table 36.10 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:

1. 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.

36.7.1 Reception

Figure 36.18 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).

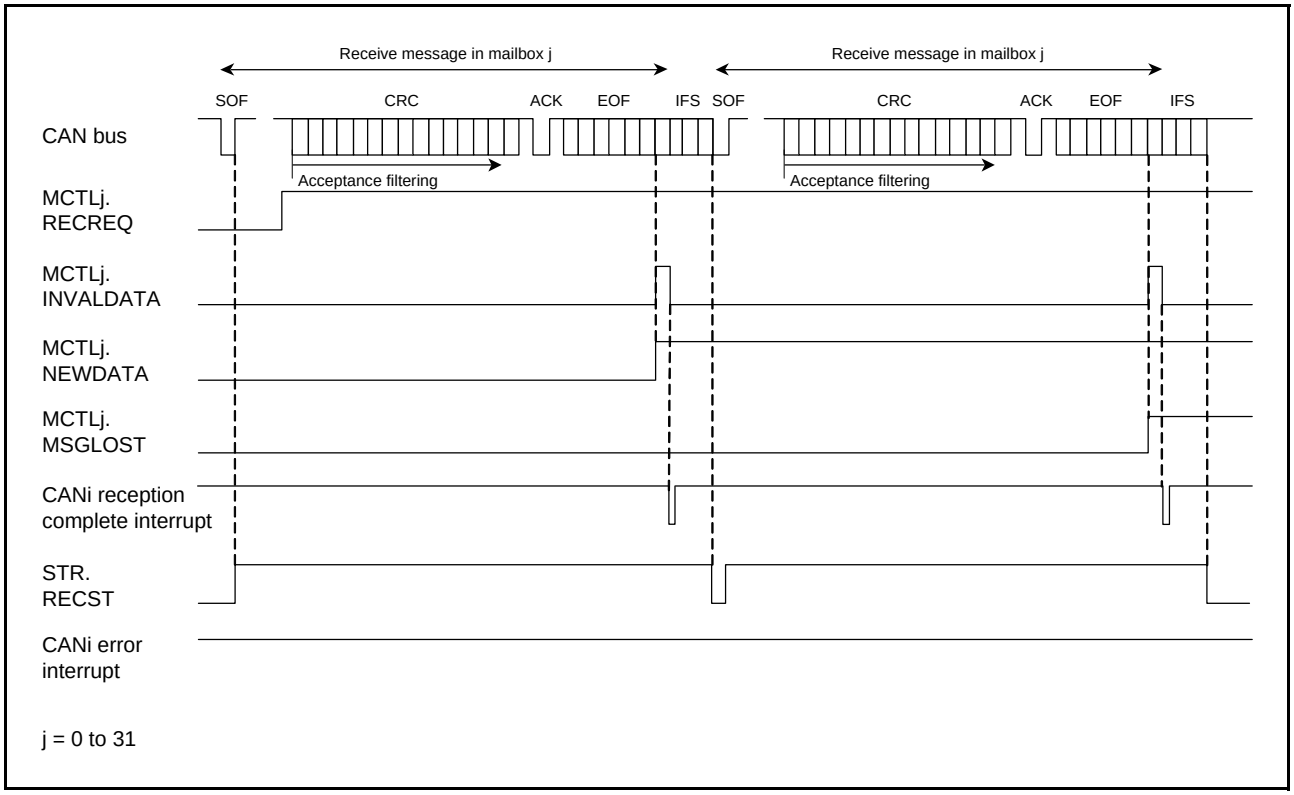


Figure 36.18 Operation Example of Data Frame Reception in Overwrite Mode

1. When an SOF is detected on the CAN bus, the RECST bit 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 bit in MCTLj for the receive mailbox is set to 1 (new message is being stored or has been stored to the mailbox). The INVALDATA bit in MCTLj is set to 1 (message is being updated) at the same time, and then the INVALDATA bit 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 bit is set to 0.
5. After reading the message from the mailbox, the NEWDATA bit 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 bit is still set to 1, the MSGLOST bit 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 36.19 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).

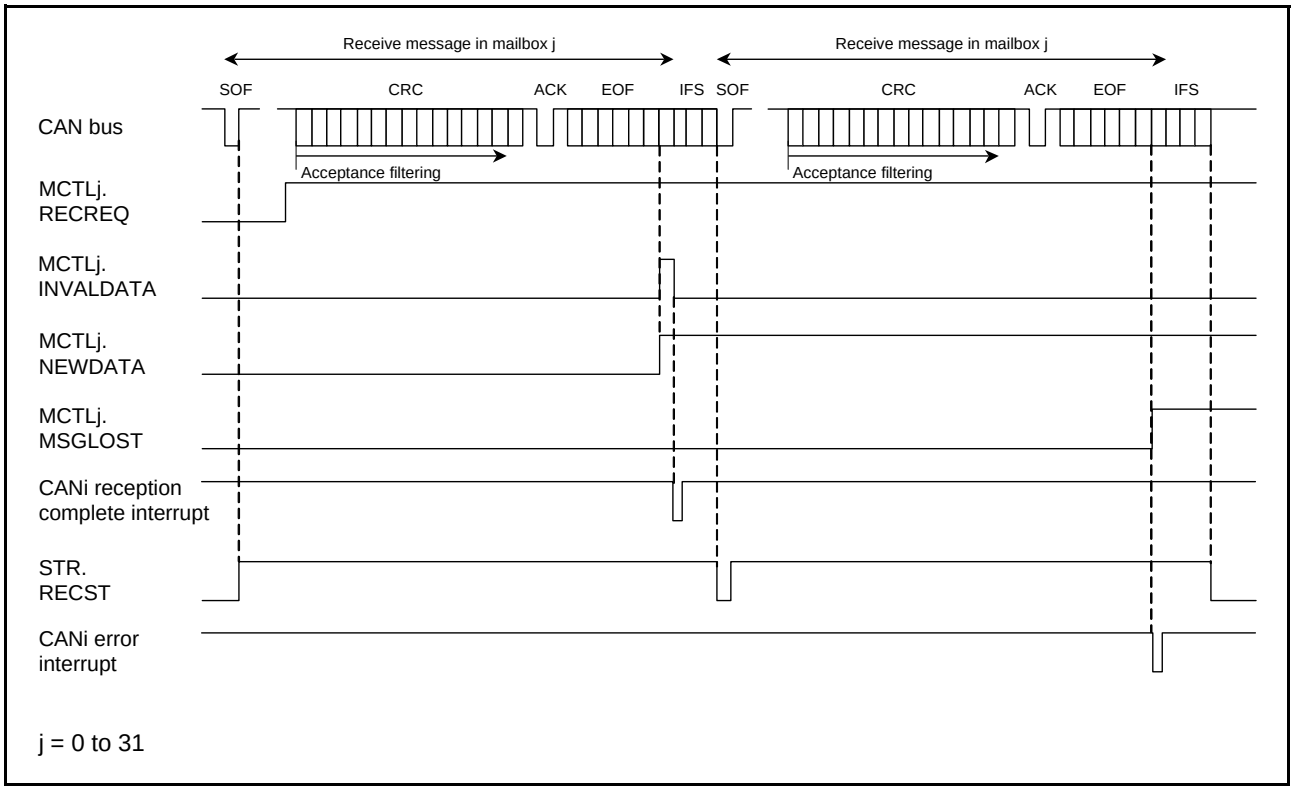


Figure 36.19 Operation Example of Data Frame Reception in Overrun Mode

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 bit in MCTLj is set to 0, the MSGLOST bit 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).

36.7.2 Transmission

Figure 36.20 shows an operation example of data frame transmission.

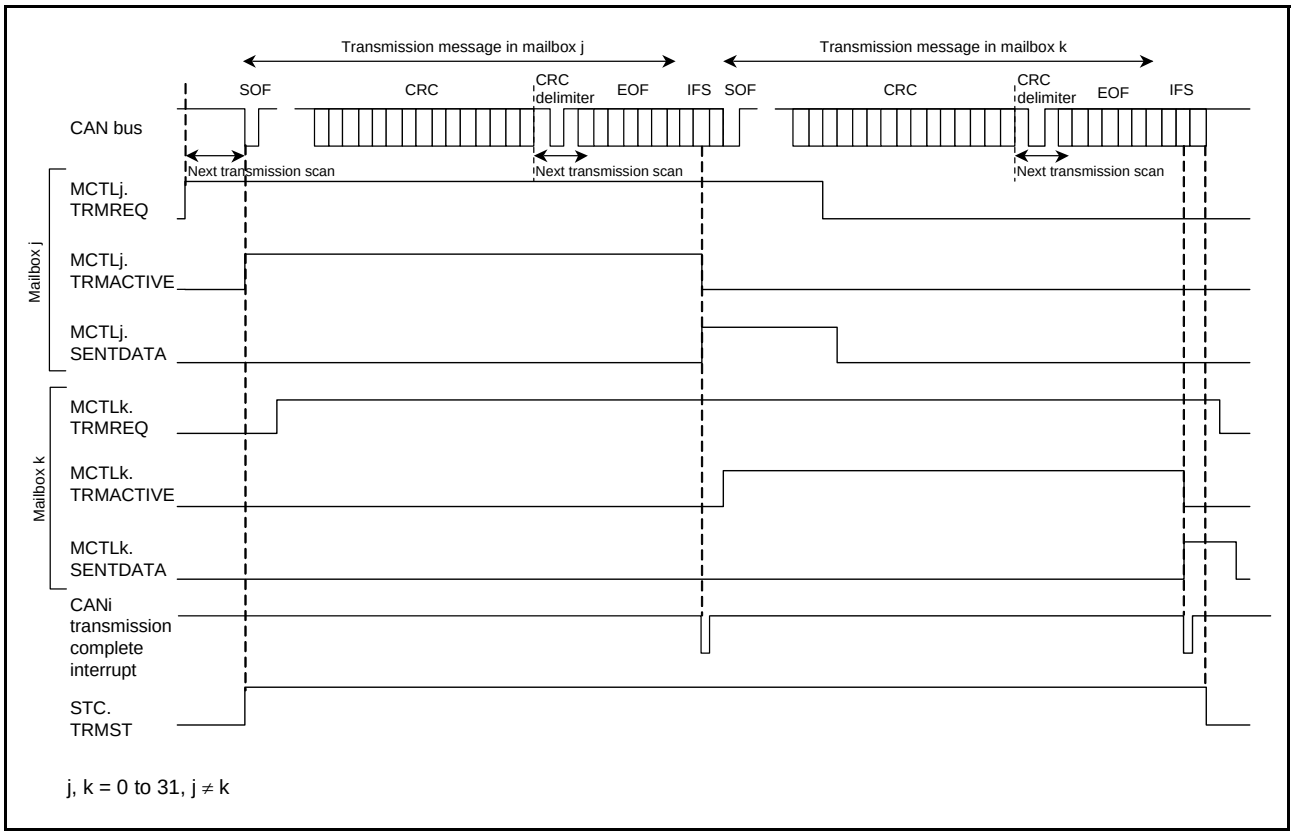


Figure 36.20 Operation Example of Data Frame Transmission

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 bit in MCTLj is set to 1 (from acceptance of transmission request to completion of transmission, or error/arbitration lost), the TRMST bit 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 bit in MCTLj is set to 1 (transmission completed) and the TRMACTIVE bit 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 bits SENDTDATA and TRMREQ to 0, then set the TRMREQ bit to 1 after checking that bits SENDTDATA and TRMREQ have been set to 0.

Note 1. If arbitration is lost after the CAN module starts transmission, the TRMACTIVE bit 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.

36.8 CAN Interrupt

The CAN module provides the following CAN interrupts for each channel. Table 36.11 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 36.11 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
		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

36.9 Usage Notes

36.9.1 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.

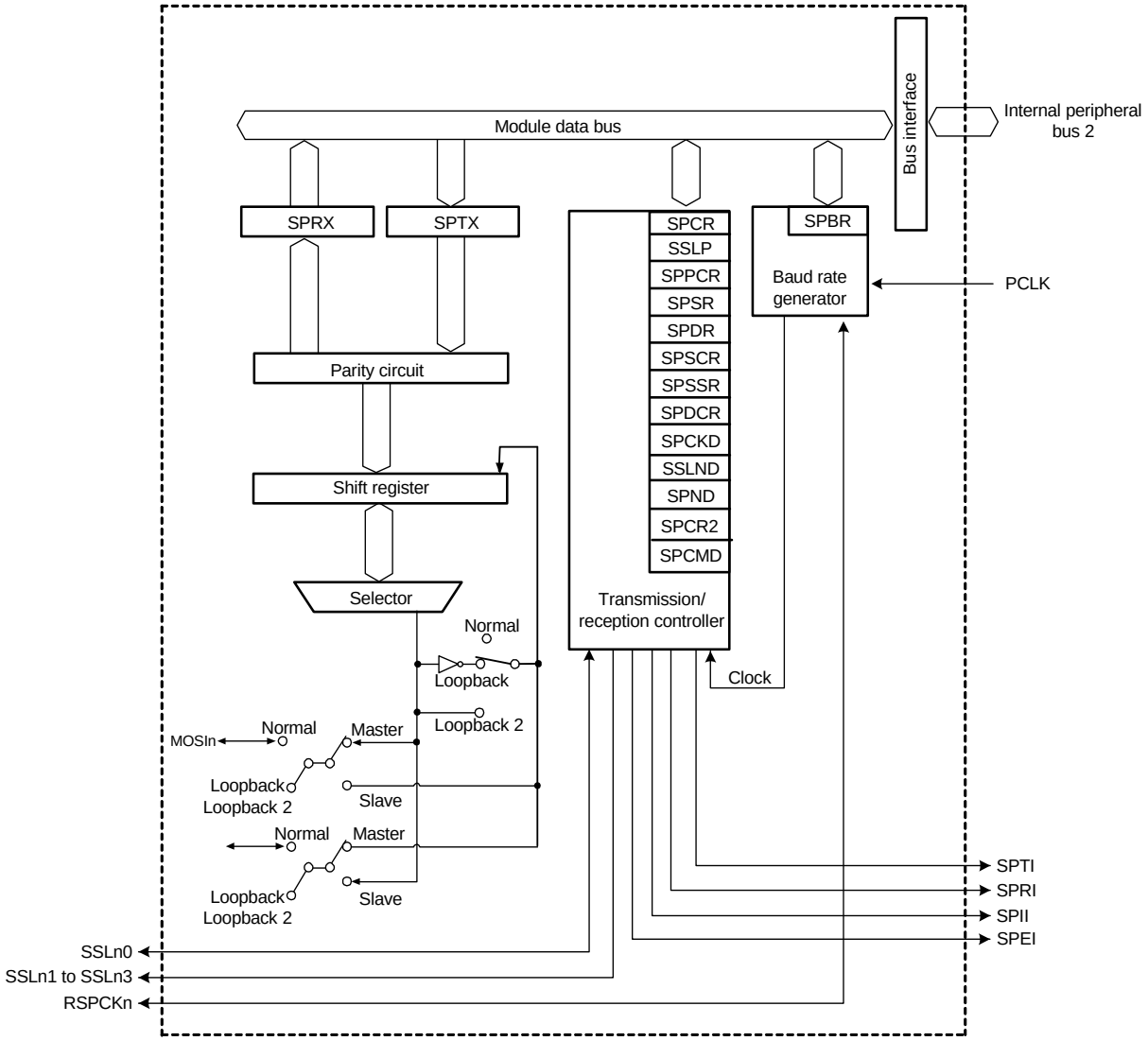
37. Serial Peripheral Interface (RSPI)

37.1 Overview

The RX63N/RX631 Group includes three independent channels 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 37.1 lists the specifications of the RSPI, and Figure 37.1 shows a block diagram of the RSPI.
Furthermore, n in this section indicates A, B, or C, and i indicates 0 to 3. Also, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

Table 37.1 Specifications of RSPI

Item	Description
Number of channels	Three channels
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 (four-wire method) or clock synchronous operation (three-wire method). • Transmit-only operation is available. • Capable of serial communications in master/slave mode • Switching of the polarity of the serial transfer clock • Switching of the phase of the serial transfer clock
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).
Buffer configuration	Double buffer configuration for the transmit/receive buffers
Error detection	• Mode fault error detection • Overrun error detection • Parity error detection
SSL control function	• Four SSL signals (SSLn0 to SSLn3) for each channel • In single-master mode, SSLn0 to SSLn3 signals are output. • In multi-master mode: SSLn0 signal for input, and SSLn1 to SSLn3 signals for either output or unused. • In slave mode: SSLn0 signal for input, and SSLn1 to SSLn3 signals 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
Interrupt sources	• Maskable interrupt sources RSPI receive interrupt (receive buffer full) RSPI transmit interrupt (transmit buffer empty) RSPI error interrupt (mode fault, overrun, parity error) RSPI idle interrupt (RSPI idle)
Others	• Function for initializing the RSPI • Loopback mode
Power consumption reducing function	Module stop state can be set.



SPCR: RSPI control register
SPCR2: RSPI control register 2
SSLP: RSPI slave select polarity register
SPPCR: RSPI pin control register
SPSR: RSPI status register
SPDR: RSPI data register
SPSCR: RSPI sequence control register
SPSSR: RSPI sequence status register
SPDCR: RSPI data control register
SPCKD: RSPI clock delay register
SSLND: RSPI slave select negate delay register
SPND: RSPI next-access delay register
SPCMD: RSPI command registers 0 to 7 (eight registers)
SPBR: RSPI bit rate register
SPTX: RSPI transmit buffer
SPRX: RSPI receive buffer
SPTI: RSPI transmit interrupt
SPRI: RSPI receive interrupt
SPII: SPI idle interrupt
SPEI: RSPI error interrupt

Figure 37.1 Block Diagram of RSPI

Table 37.2 lists the input and output pins used in the RSPI.
The RSPI automatically switches the input/output direction of the SSLn0 pin (n = A, B, or C). SSLn0 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 RSPCKn, MOSIn, and MISOn (n = A, B, or C) are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLn0 pin (see section 37.3.2, Controlling RSPI Pins).

Table 37.2 RSPI Pin Configuration

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O pin
	MOSIA	I/O	Master transmit data I/O pin
	MISOA	I/O	Slave transmit data I/O pin
	SSLA0	I/O	Slave selection I/O pin
	SSLA1	Output	Slave selection output pin
	SSLA2	Output	Slave selection output pin
	SSLA3	Output	Slave selection output pin
RSPI1	RSPCKB	I/O	Clock I/O pin
	MOSIB	I/O	Master transmit data I/O pin
	MISOB	I/O	Slave transmit data I/O pin
	SSLB0	I/O	Slave selection I/O pin
	SSLB1	Output	Slave selection output pin
	SSLB2	Output	Slave selection output pin
	SSLB3	Output	Slave selection output pin
RSPI2	RSPCKC	I/O	Clock I/O pin
	MOSIC	I/O	Master transmit data I/O pin
	MISOC	I/O	Slave transmit data I/O pin
	SSLC0	I/O	Slave selection I/O pin
	SSLC1	Output	Slave selection output pin
	SSLC2	Output	Slave selection output pin
	SSLC3	Output	Slave selection output pin

37.2 Register Descriptions

37.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h, RSPI1.SPCR 0008 83A0h, RSPI2.SPCR 0008 83C0h

	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 (four-wire method) 1: Clock synchronous operation (three-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	RSPI Transmit Interrupt Enable	0: Disables the generation of RSPI transmit interrupt requests 1: Enables the generation of RSPI transmit 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 Interrupt Enable	0: Disables the generation of RSPI receive interrupt requests 1: Enables the generation of RSPI receive interrupt requests	R/W

If the SPCR.MSTR, SPCR.MODFEN, and SPCR.TXMD bits are changed while the SPCR.SPE bit is 1, subsequent operations cannot be guaranteed.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (four-wire method) or clock synchronous operation (three-wire method).
The SSLn0 to SSLn3 pins are not used in clock synchronous operation. The three pins RSPCKn, MOSIn, and MISOn 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 is not guaranteed 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 (see section 37.3.6, 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 (see section 37.3.8, Error Detection). In addition, the RSPI determines the input/output direction of the SSLn0 to SSLn3 pins based on combinations of the MODFEN and MSTR bits (see section 37.3.2, 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 RSPCKn, MOSIn, MISOn, and SSLn0 to SSLn3.

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 (see section 37.3.8, Error Detection).

SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function. When the SPSR.MODF bit is 1, the SPE bit cannot be set to 1. For details, refer to section 37.3.8, 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 37.3.9, Initializing RSPI. Furthermore, an RSPI transmission interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

37.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 0008 8381h, RSPI1.SSLP 0008 83A1h, RSPI2.SSLP 0008 83C1h

	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 0-active 1: SSL0 signal is 1-active	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is 0-active 1: SSL1 signal is 1-active	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: SSL2 signal is 0-active 1: SSL2 signal is 1-active	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is 0-active 1: SSL3 signal is 1-active	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 are not guaranteed.

37.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 0008 8382h, RSPI1.SPPCR 0008 83A2h, RSPI2.SPPCR 0008 83C2h

	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 (reversed transmit data = receive data)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (transmit data = receive data)	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: MOSI Idle fixed value equals 0 1: MOSI Idle fixed value equals 1	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 are not guaranteed.

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 MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects (reverses) 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 MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn 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 MOSIn 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 MOSIn 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 MOSIn pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIn pin.

37.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h, RSPI1.SPSR 0008 83A3h, RSPI2.SPSR 0008 83C3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PERF	MODF	IDLNF	OVRF
Value after reset:	x	0	x	0	0	0	0	0

x: Undefined

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	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Note 1. Only 0 can be written to clear the flag after reading 1.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer holds data that has not yet been read

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then writes the value 0 to the OVRF flag.

IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

<Master mode>

- Even one condition among the following clearing conditions for master mode is not satisfied

<Slave mode>

- The SPCR.SPE bit is 1 (RSPI function is enabled)

[Clearing condition]

<Master mode>

1. The SPCR.SPE bit is 0 (RSPI is initialized)
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)

The flag is cleared to 0 when the above first clearing condition is satisfied or all of the second to fourth clearing conditions are satisfied.

<Slave mode>

- The SPCR.SPE bit is 0 (RSPI is initialized)

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 SSLni 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 SSLnni 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 SSLni signal is determined by the SSLP.SSLiP bit (SSL signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then writes the value 0 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 writes the value 0 to the PERF flag

37.2.5 RSPI Data Register (SPDR)

Address(es): RSPI0.SPDR 0008 8384h, RSPI1.SPDR 0008 83A4h, RSPI2.SPDR 0008 83C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
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
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
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. The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 37.2 shows the Configuration of SPDR

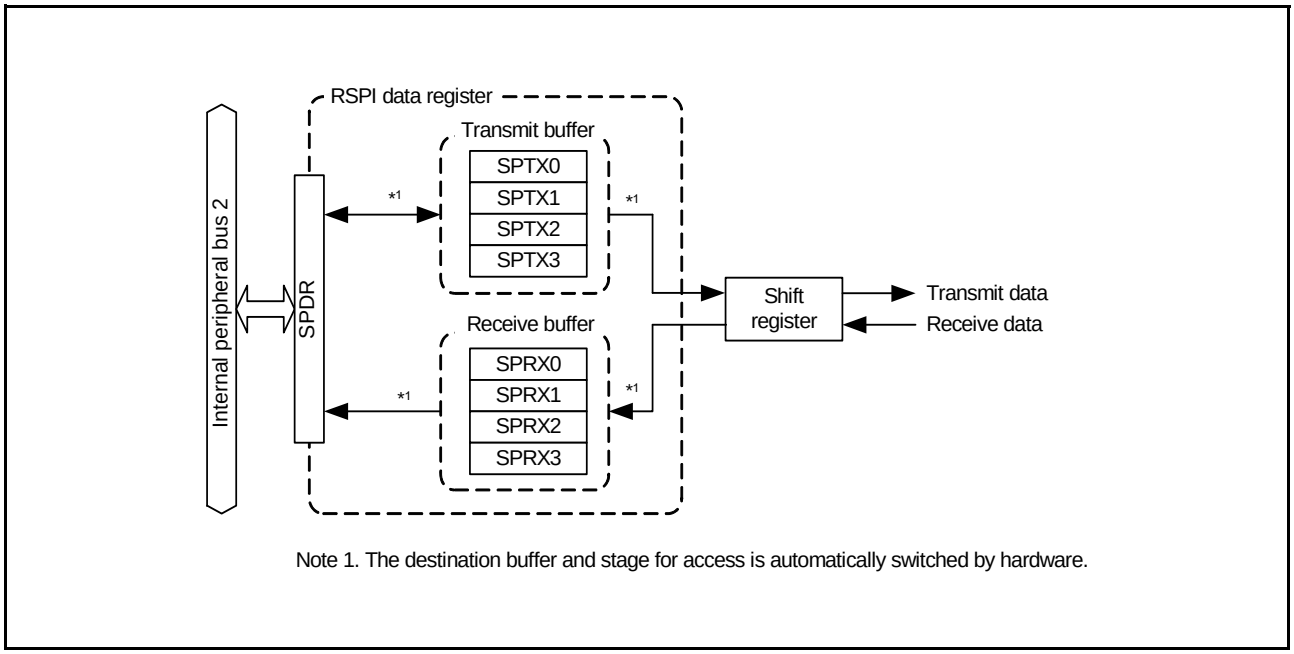


Figure 37.2 Configuration of SPDR

The transmit and receive buffers have four stages each. The number of stages to be used is selectable by the frame-number setting 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 (SPTX0 to SPTX3) 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 nine bits, the SPTXn[31:9] bits are stored in SPRXn[31:9] (and received data are stored in the SPTXn[8:0] 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 four-byte address space for SPDR. Furthermore, the unit of access for SPDR is selected by the long-word or word access setting bit in the RSPI data control register (SPDCR.SPLW).
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

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 37.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.

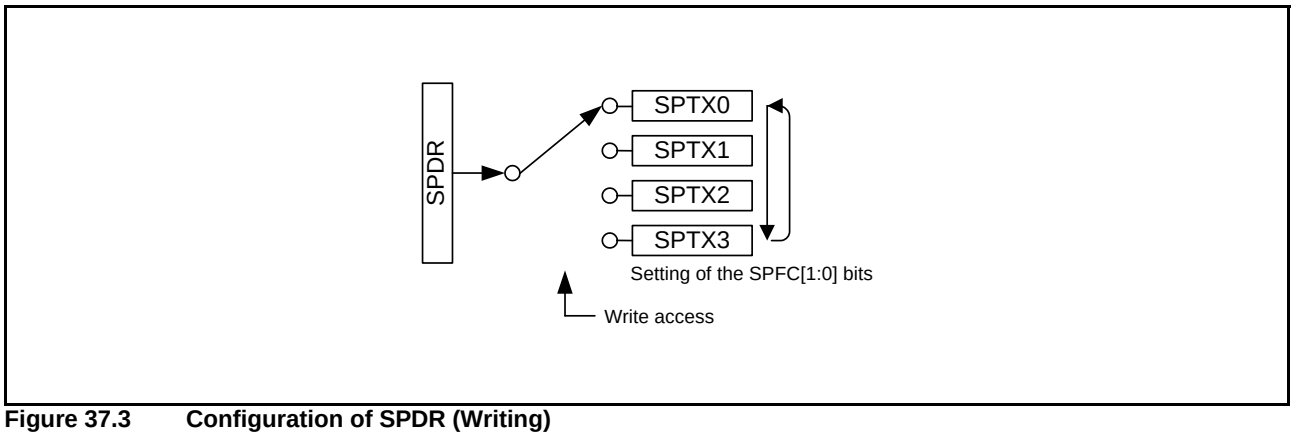


Figure 37.3 Configuration of SPDR (Writing)

The sequence for switching the transmit buffer write pointer differs with the setting of the frame-number setting 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 SPTX03.
  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 transmission buffer after generation of the RSPI transmission interrupt, write the number of frames set by the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Writing to the transmission buffer before the next RSPI transmission interrupt breaks off write access to the SPDR.

(b) Reading

SPDR can be read to read out the value of a receive buffer (SPRXn; n = 0 to 3) or a transmit buffer (SPTXn; n = 0 to 3). The setting of the RSPI transmit/receive data selection bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.
The structure of the SPDR when it is read includes two independent pointers (receive buffer read pointer and transmit buffer read pointer). Reading SPDR causes automatic updating of the pointer so that it indicates the next stage of the buffer.
Figure 37.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

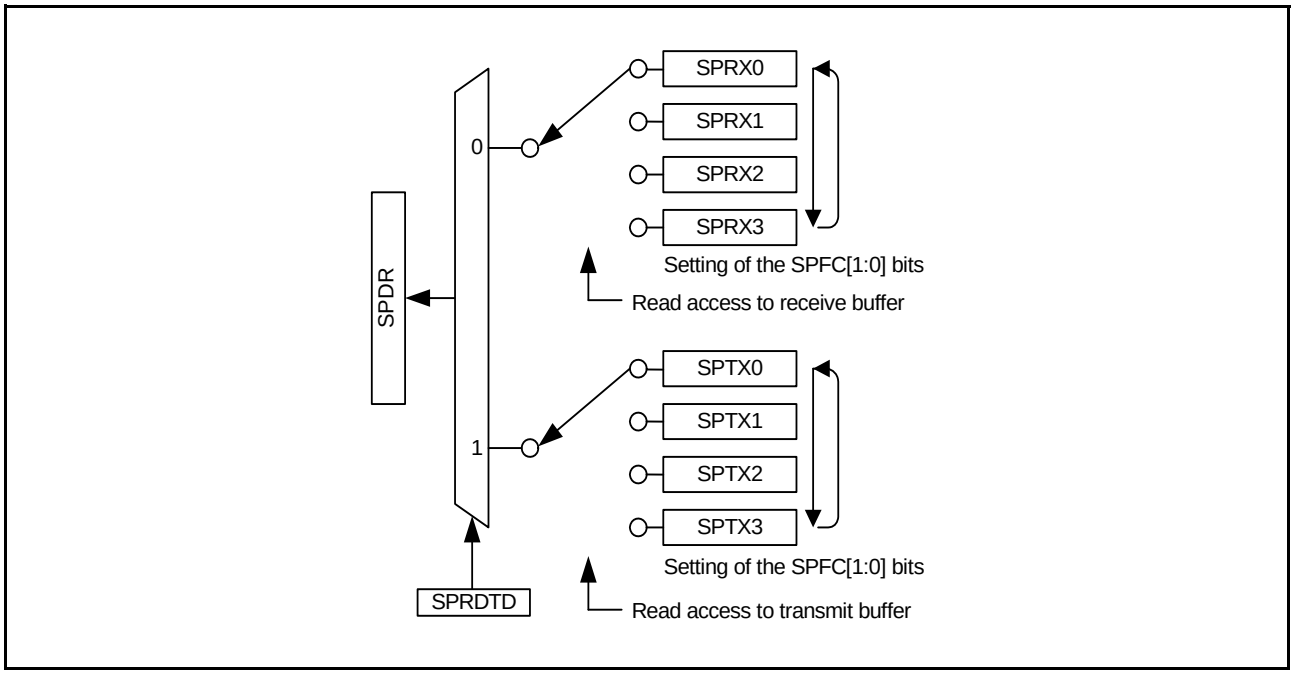


Figure 37.4 Configuration of SPDR (Reading)

The sequence for switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. When reading is from the transmit buffer, the value before that most recently written is read. Furthermore, only the read pointer for the buffer that is currently selected for reading by the setting of the RSPI transmit/receive data selection bit in the RSPI data control register (SPDCR.SPRDTD) is updated, and the state of the read pointer for the other buffer is preserved.

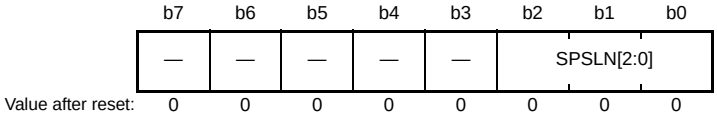
When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

If the transmission buffer is read in the interval after writing of the number of frames of data for transmission specified in the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR) and generation of the RSPI transmission interrupt, and before the next RSPI transmission interrupt, all bits of the value read out are 0.

The SPTXn buffer-reading pointer is cleared when the number of frames of data for transmission specified in the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR) are written after generation of the RSPI transmission interrupt.

37.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 0008 8388h, RSPI1.SPSCR 0008 83A8h, RSPI2.SPSCR 0008 83C8h



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→… SPCMD0 to SPCMD7 to be referenced and the order in which they are referenced are 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 always references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

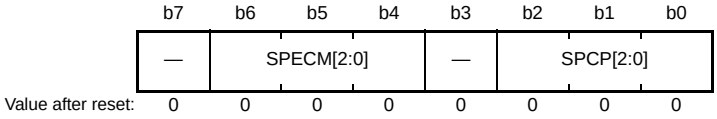
SPSCR sets the sequence length when the RSPI operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

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 always referred to.

37.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 0008 8389h, RSPI1.SPSSR 0008 83A9h, RSPI2.SPSSR 0008 83C9h



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, see section 37.3.10.1, 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 bits 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, see section 37.3.8, Error Detection. For the RSPI's sequence control, see section 37.3.10.1, Master Mode Operation.

37.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah, RSPI1.SPBR 0008 83AAh, RSPI2.SPBR 0008 83CAh

	b7	b6	b5	b4	b3	b2	b1	b0
	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
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 cannot be guaranteed.

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).

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2\times(n+1)\,2^{N}}$$

Table 37.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.

Table 37.3 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
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps
5	2	48	667 kbps	750 kbps	833 kbps	1.04 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps

37.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh, RSPI1.SPDCR 0008 83ABh, RSPI2.SPDCR 0008 83CBh

	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 Selection	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 reception interrupts, and start of transmission or generation of RSPI transmission interrupts. Table 37.4 show 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 are not guaranteed.

SPRDTD Bit (RSPI Receive/Transmit Data Selection)

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 RSPI transmission interrupt.

SPLW Bit (RSPI Longword Access/Word Access Specification)

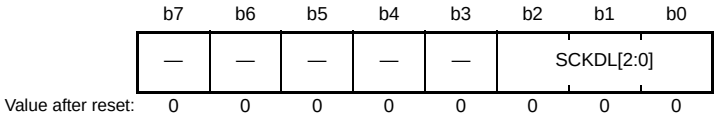
The SPLW bit specifies the access width for SPDR. Access to SPDR is in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length specification bits) to 8 to 16 bits. When 20, 24, or 32 bits is specified, operation is not guaranteed.

Table 37.4 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 Receive Buffer Full Interrupt Occurs or Transmit Buffer Holding Data is Recognized
1-1	000	00	1	1
1-2	000	01	2	2
1-3	000	10	3	3
1-4	000	11	4	4
2-1	001	01	2	2
2-2	001	11	4	4
3	010	10	3	3
4	011	11	4	4
5	100	00	5	1
6	101	00	6	1
7	110	00	7	1
8	111	00	8	1

37.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 0008 838Ch, RSPI1.SPCKD 0008 83ACh, RSPI2.SPCKD 0008 83CCh



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 SSLni signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the contents of SPCKD are changed by the CPU while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations cannot be guaranteed.

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.

37.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 0008 838Dh, RSPI1.SSLND 0008 83ADh, RSPI2.SSLND 0008 83CDh

	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 SSLni signal during a serial transfer by the RSPI in master mode. If the contents of SSLND are changed by the CPU while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations cannot be guaranteed.

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.

37.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 0008 838Eh, RSPI1.SPND 0008 83AEh, RSPI2.SPND 0008 83CEh

	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]	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 SSLni 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 cannot be guaranteed.

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.

37.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 0008 838Fh, RSPI1.SPCR2 0008 83AFh, RSPI2.SPCR2 0008 83CFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	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-Testing	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the SPPE bit or SPOE bit in SPCR2 is changed while the SPCR.SPE bit is 1, subsequent operations cannot be guaranteed.

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 cleared to 0.

PTE Bit (Parity Self-Testing)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

37.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)

Address(es): RSPI0.SPCMD0 0008 8390h, RSPI0.SPCMD1 0008 8392h, RSPI0.SPCMD2 0008 8394h, RSPI0.SPCMD3 0008 8396h, RSPI0.SPCMD4 0008 8398h, RSPI0.SPCMD5 0008 839Ah, RSPI0.SPCMD6 0008 839Ch, RSPI0.SPCMD7 0008 839Eh, RSPI1.SPCMD0 0008 83B0h, RSPI1.SPCMD1 0008 83B2h, RSPI1.SPCMD2 0008 83B4h, RSPI1.SPCMD3 0008 83B6h, RSPI1.SPCMD4 0008 83B8h, RSPI1.SPCMD5 0008 83BAh, RSPI1.SPCMD6 0008 83BCh, RSPI1.SPCMD7 0008 83BEh RSPI2.SPCMD0 0008 83D0h, RSPI2.SPCMD1 0008 83D2h, RSPI2.SPCMD2 0008 83D4h, RSPI2.SPCMD3 0008 83D6h, RSPI2.SPCMD4 0008 83D8h, RSPI2.SPCMD5 0008 83DAh, RSPI2.SPCMD6 0008 83DCh, RSPI2.SPCMD7 0008 83DEh

	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

Bit	Symbol	Bit Name	Description	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 cannot be guaranteed.

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 (see section 37.2.8, 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 permits the 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 SSLni signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLni signal. When an SSLni 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 SLL signals in the negated state (as the SSLn0 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 SSLni 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.
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.

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 SSLni signal inactive until the RSPI enables the SSLni 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 SSLni 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 SSLni 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.

37.3 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.

37.3.1 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 37.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 37.5 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
RSPCKn signal	Input	Output	Output/Hi-Z	Input	Output
MOSIn signal	Input	Output	Output/Hi-Z	Input	Output
MISOn signal	Output/Hi-Z	Input	Input	Output	Input
SSLn0 signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLn1 to SSLn3 signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity modification 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	Two	Two	Two	Two
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
Transfer data length	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 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	Transmit buffer is written to at generation of a transmit buffer empty interrupt request	RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported	Supported	Supported	Supported	Supported
Receive buffer full detection	Supported*2	Supported*2	Supported*2	Supported*2	Supported*2
Overrun error detection	Supported*2	Supported*2	Supported*2	Supported*2	Supported*2
Parity error detection	Supported*2,*3	Supported*2,*3	Supported*2,*3	Supported*2,*3	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.

37.3.2 Controlling RSPI Pins

According to the MSTR, MODFEN, and SPMS bits in SPCR and the ODRn.Bi bit for I/O port, the RSPI can switch pin states. Table 37.6 lists the relationship between pin states and bit settings. The I/O port settings should follow this relationship.

Table 37.6 Relationship between Pin States and Bit Settings

Mode	Pin	Pin State*2	
		ODRn.Bi = 0	ODRn.Bi = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOn	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn	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 SSLn0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLn0 is at the non-active level or the SPCR.SPE bit is cleared (= 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 37.7.

Table 37.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSIn Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Always low
1	1	Always high

37.3.3 RSPI System Configuration Examples

37.3.3.1 Single Master/Single Slave (with This LSI Acting as Master)

Figure 37.5 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a master. In the single-master/single-slave configuration, the SSLn0 to SSLn3 output of this LSI (master) are not used. The SSL input of the RSPI slave is fixed to the low level, and the RSPI slave is always maintained in a select state.*1
This LSI (master) always drives the RSPCKn and MOSIn. The RSPI slave always 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 SSLni output of this LSI should be connected to the SSL input of the slave device.

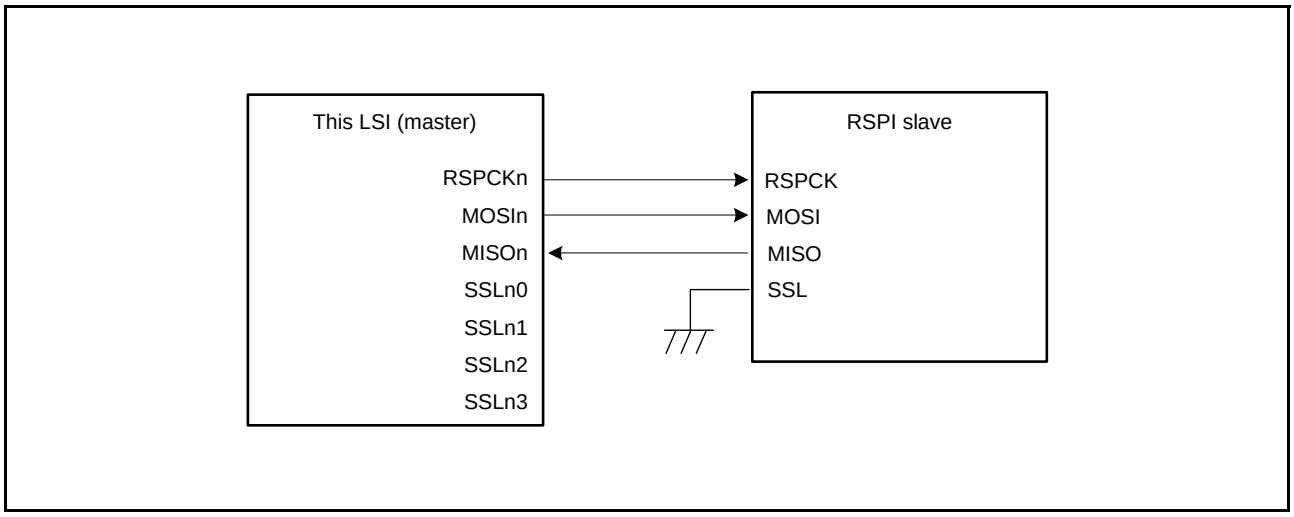


Figure 37.5 Single-Master/Single-Slave Configuration Example (This LSI = Master)

37.3.3.2 Single Master/Single Slave (with This LSI Acting as Slave)

Figure 37.6 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSLn0 pin is used as SSL input. The RSPI master always drives the RSPCK and MOSI. This LSI (slave) always drives the MISOn.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLn0 input of this LSI (slave) is fixed to the low level, this LSI (slave) is always maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 37.7).

Note 1. When SSLn0 is at the non-active level, the pin state is Hi-Z.

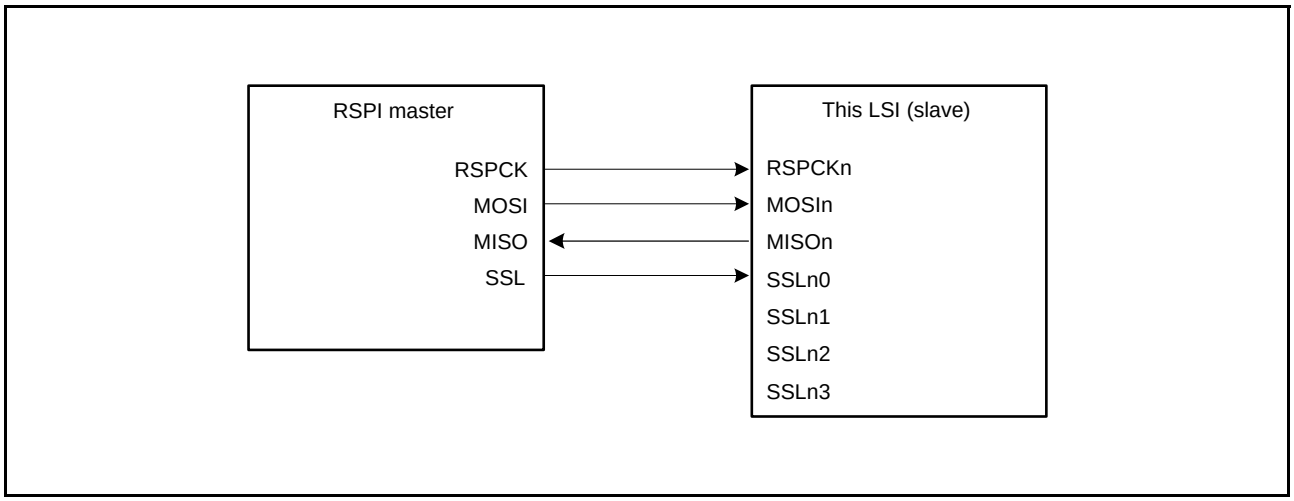


Figure 37.6 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 0)

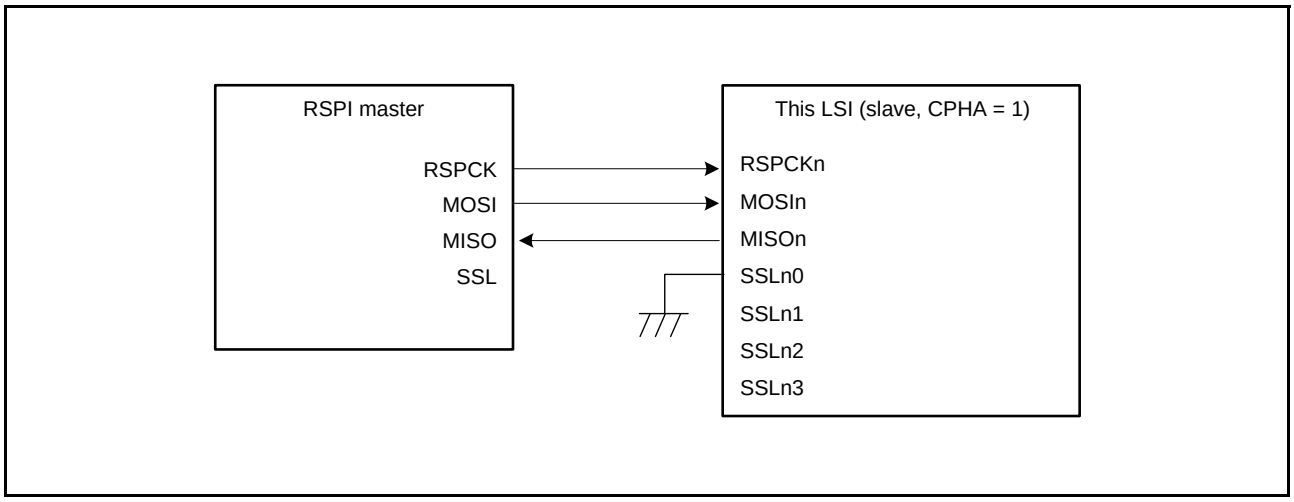


Figure 37.7 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

37.3.3.3 Single Master/Multi-Slave (with This LSI Acting as Master)

Figure 37.8 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of Figure 37.8, the RSPI system is comprised of this LSI (master) and four slaves (RSPI slave 0 to RSPI slave 3).

The RSPCKn and MOSIn outputs of this LSI (master) are connected to the RSPCK and MOSI inputs of RSPI slave 0 to RSPI slave 3. The MISO outputs of RSPI slave 0 to RSPI slave 3 are all connected to the MISOn input of this LSI (master). SSLn0 to SSLn3 outputs of this LSI (master) are connected to the SSL inputs of RSPI slave 0 to RSPI slave 3, respectively.

This LSI (master) always drives RSPCK, MOSI, and SSLn0 to SSLn3. Of the RSPI slave 0 to RSPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

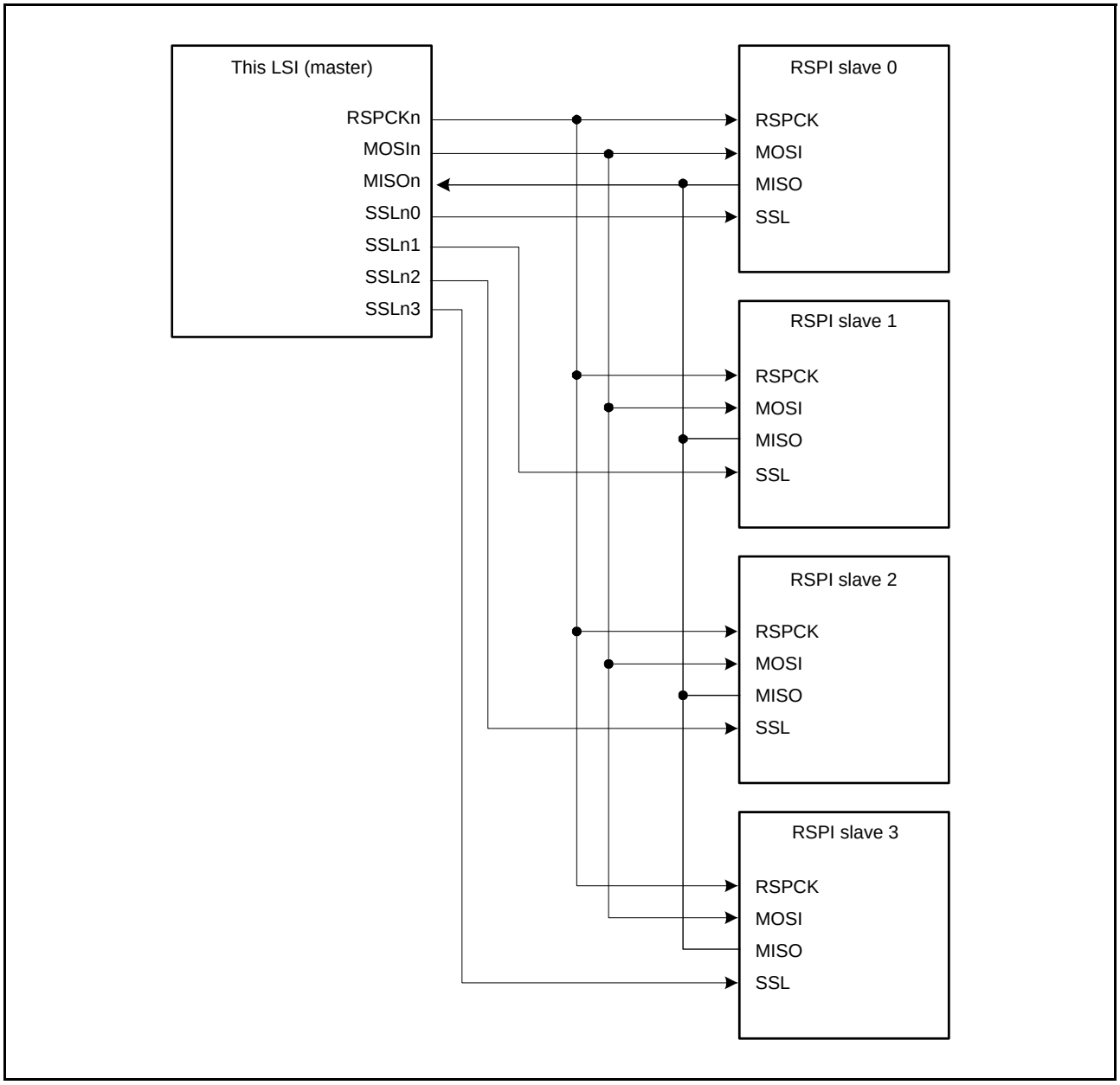


Figure 37.8 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

37.3.3.4 Single Master/Multi-Slave (with This LSI Acting as Slave)

Figure 37.9 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a slave. In the example of Figure 37.9, the RSPI system is comprised of an RSPI master and two LSIs (slave X and slave Y).
The RSPCK and MOSI outputs of the RSPI master are connected to the RSPCKn and MOSIn inputs of the LSIs (slave X and slave Y). The MISOn outputs of the LSIs (slave X and slave Y) are all connected to the MISO input of the RSPI master. SSLX and SSLY outputs of the RSPI master are connected to the SSLn0 inputs of the LSIs (slave X and slave Y), respectively.
The RSPI master always drives RSPCK, MOSI, SSLX, and SSLY. Of the LSIs (slave X and slave Y), the slave that receives low-level input into the SSLn0 input drives MISOn.

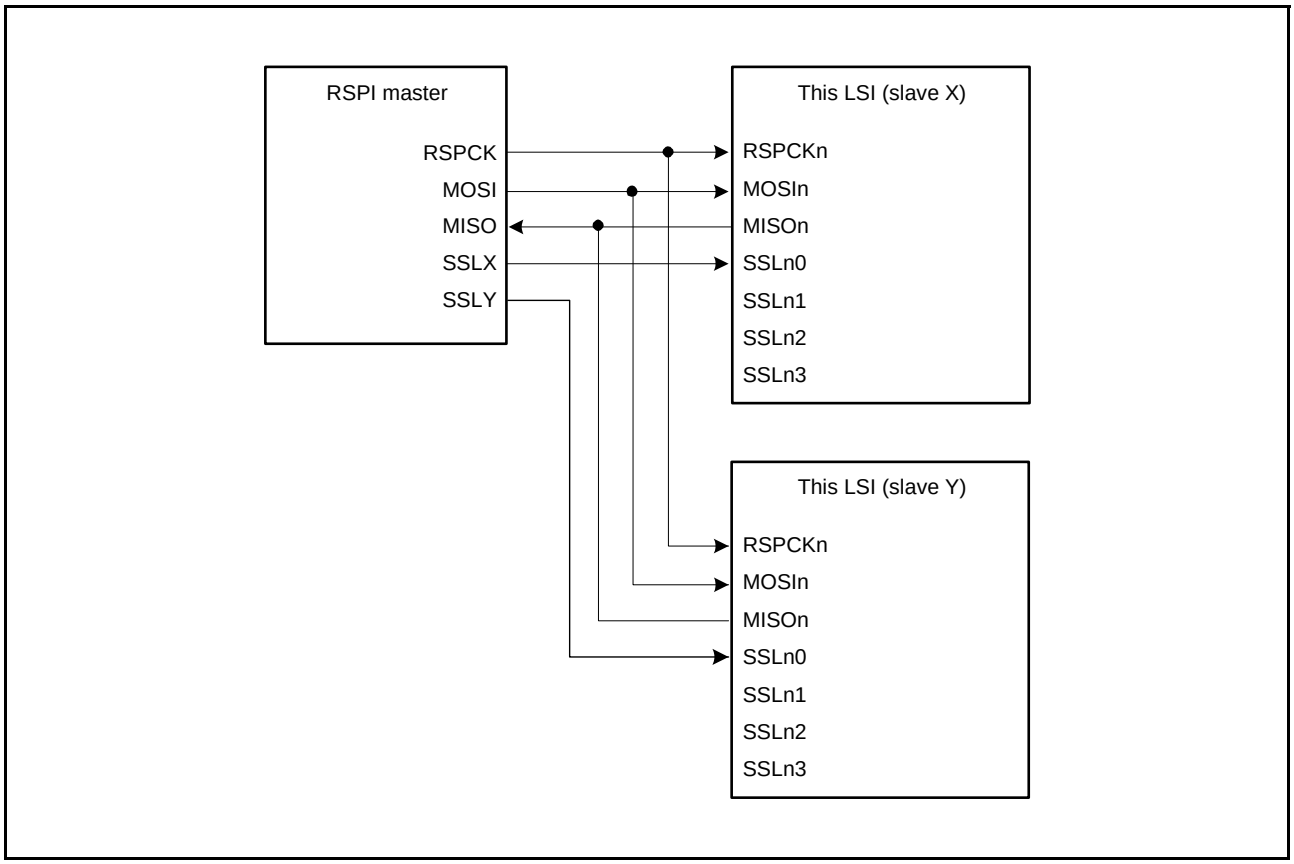


Figure 37.9 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

37.3.3.5 Multi-Master/Multi-Slave (with This LSI Acting as Master)

Figure 37.10 shows a multi-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of Figure 37.10, the RSPI system is comprised of two LSIs (master X and master Y) and two RSPI slaves (RSPI slave 1 and RSPI slave 2).

The RSPCKn and MOSIn outputs of the LSIs (master X and master Y) are connected to the RSPCK and MOSI inputs of RSPI slaves 1 and 2. The MISO outputs of RSPI slaves 1 and 2 are connected to the MISOn inputs of the LSIs (master X and master Y). Any generic port Y output from this LSI (master X) is connected to the SSLn0 input of this LSI (master Y). Any generic port X output of this LSI (master Y) is connected to the SSLn0 input of this LSI (master X). The SSLn1 and SSLn2 outputs of the LSIs (master X and master Y) are connected to the SSLn inputs of the RSPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of this LSI is not required.

This LSI drives RSPCKn, MOSIn, SSLn1, and SSLn2 when the SSLn0 input level is high. When the SSLn0 input level is Low, this LSI detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the RSPI bus right to the other master. Of the RSPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

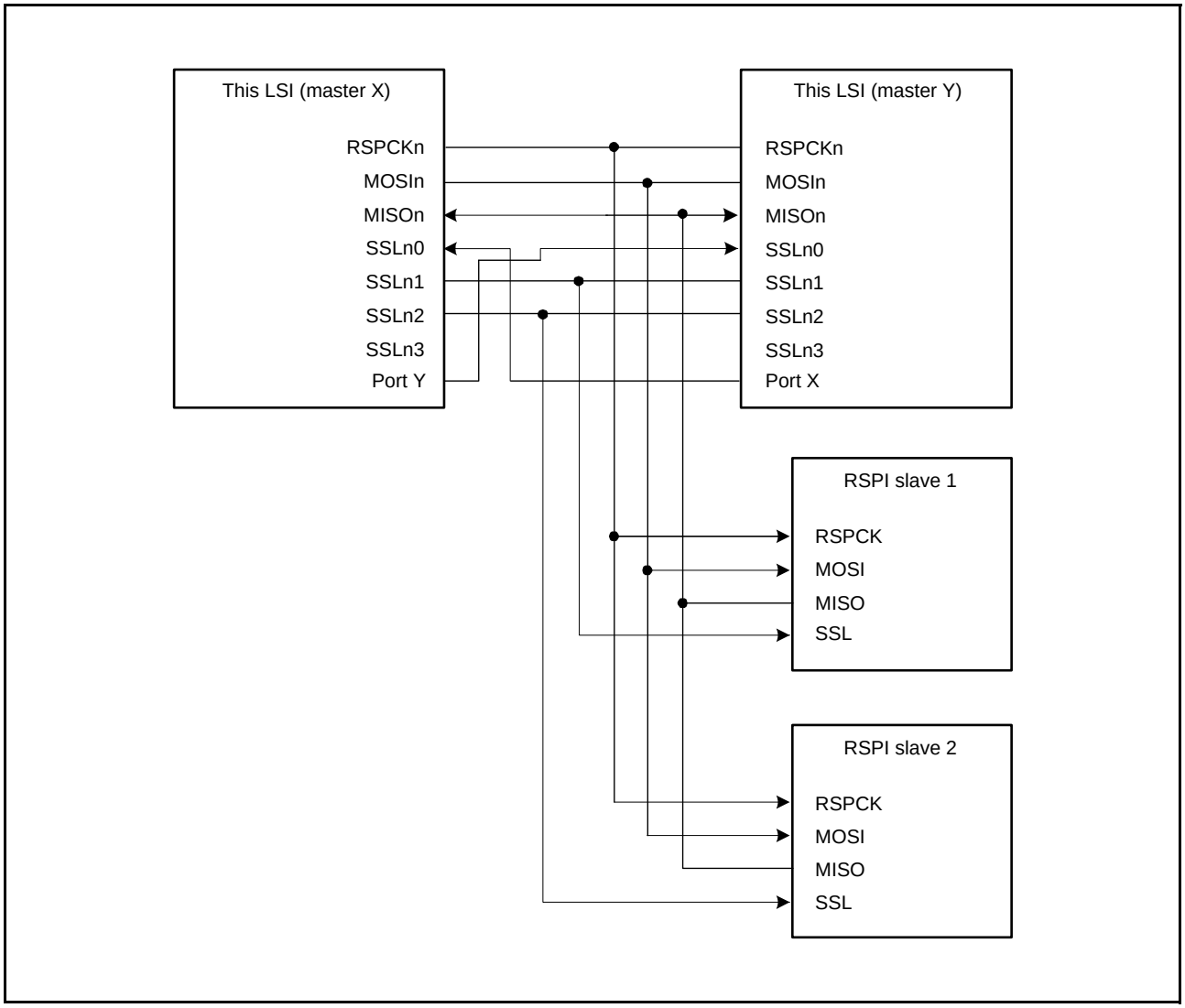


Figure 37.10 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

37.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Master)

Figure 37.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this LSI is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLn0 to SSLn3 of this LSI (master) are not used.
This LSI (master) always drives the RSPCKn and MOSIn. The RSPI slave always drives the MISO.

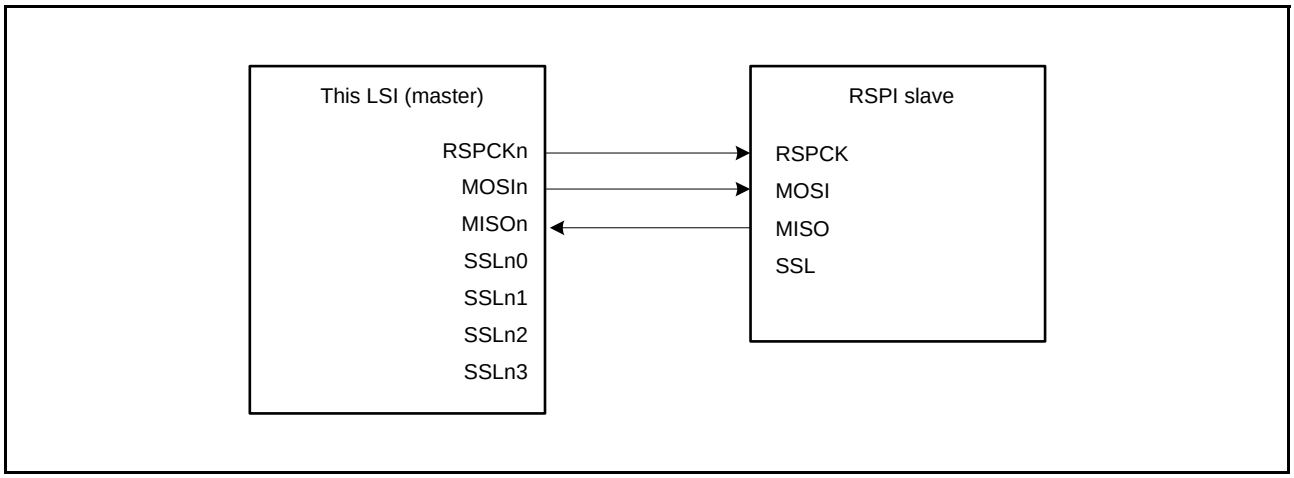


Figure 37.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This LSI = Master)

37.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Slave)

Figure 37.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave (clock synchronous operation), this LSI (slave) always drives the MISOn and the RSPI master always drives the RSPCK and MOSI. In addition, SSLn0 to SSLn3 of this LSI (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this LSI (slave) can execute serial transfer.

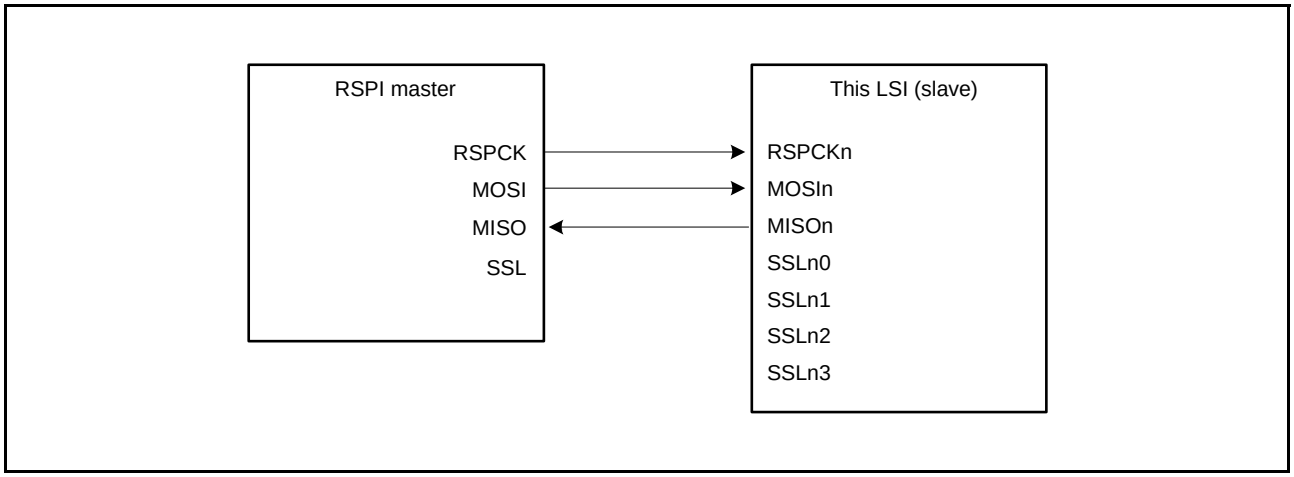


Figure 37.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This LSI = Slave, CPHA = 1)

37.3.4 Data Format

The RSPI's data format depends on the settings in the RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit of RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit of 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 data length setting bits in the RSPI command register m (SPCMDm.SPB[3:0]).

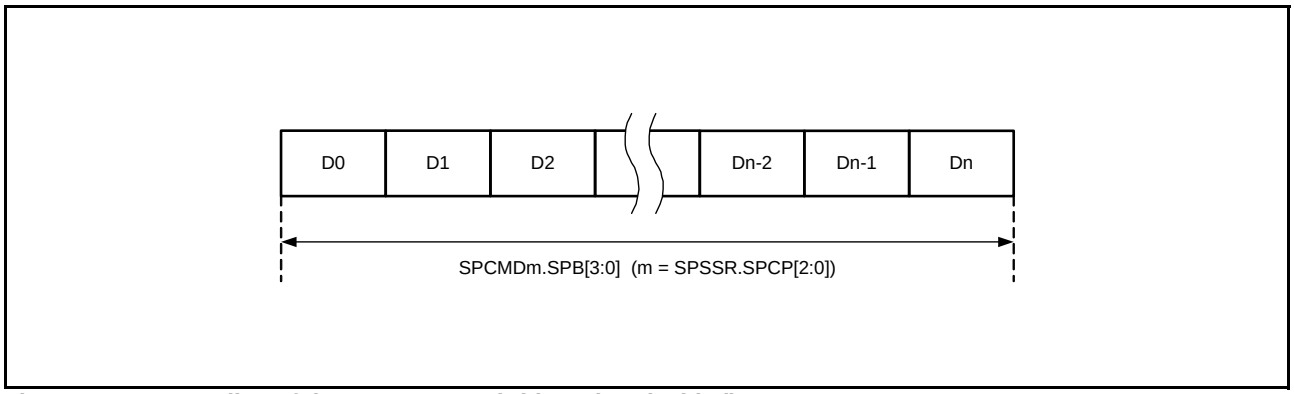


Figure 37.13 Outline of the Data Format (with Parity Disabled)

(b) With Parity Enabled

When parity is enabled, transmission or reception of data also proceeds with the length in bits selected in the SPCMDm.SPB[3:0] bits. In this case, however, the last bit is a parity bit.

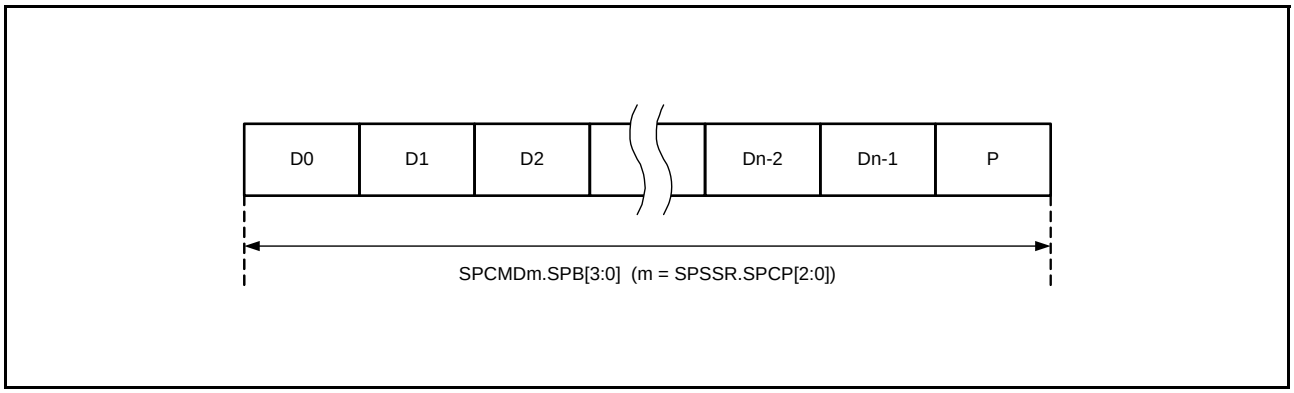


Figure 37.14 Outline of the Data Format (with Parity Enabled)

37.3.4.1 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 37.15 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.

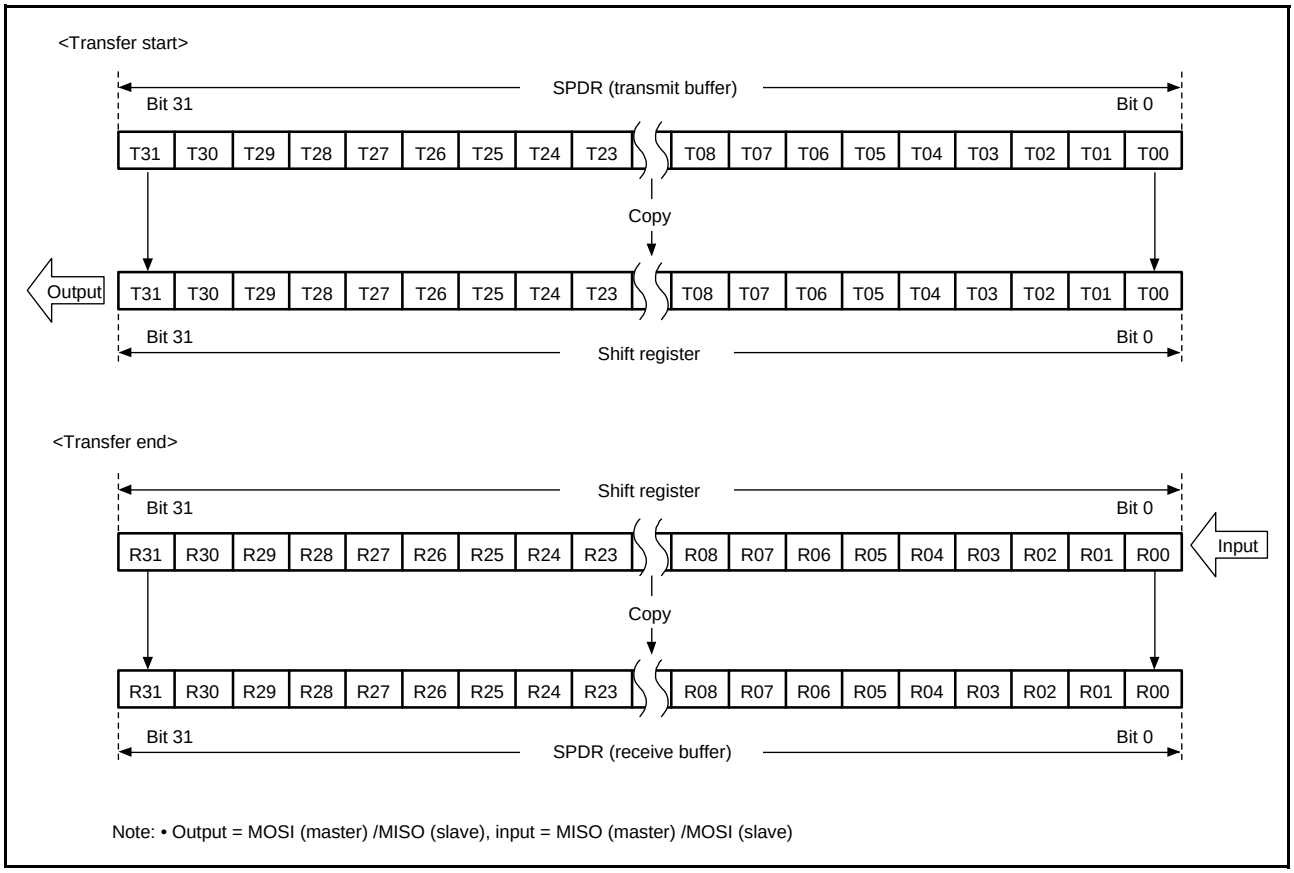


Figure 37.15 MSB-First Transfer (32-Bit Data, Parity Disabled)

(2) MSB-First Transfer (24-Bit Data)

Figure 37.16 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 eight bits of the transmit buffer are stored in the higher-order eight 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 eight bits of the receive buffer.

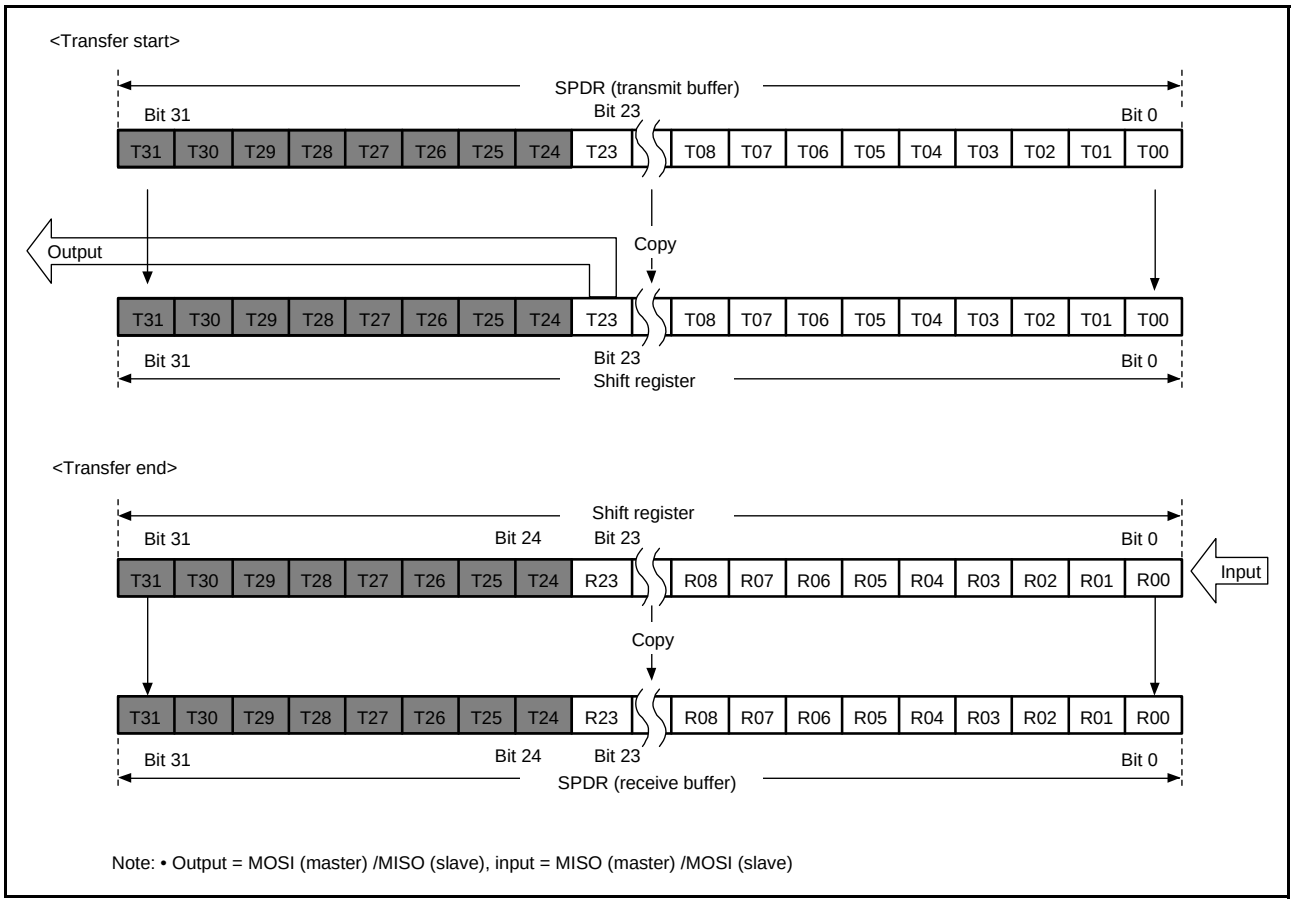


Figure 37.16 MSB-First Transfer (24-Bit Data, Parity Disabled)

(3) LSB-First Transfer (32-Bit Data)

Figure 37.17 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.

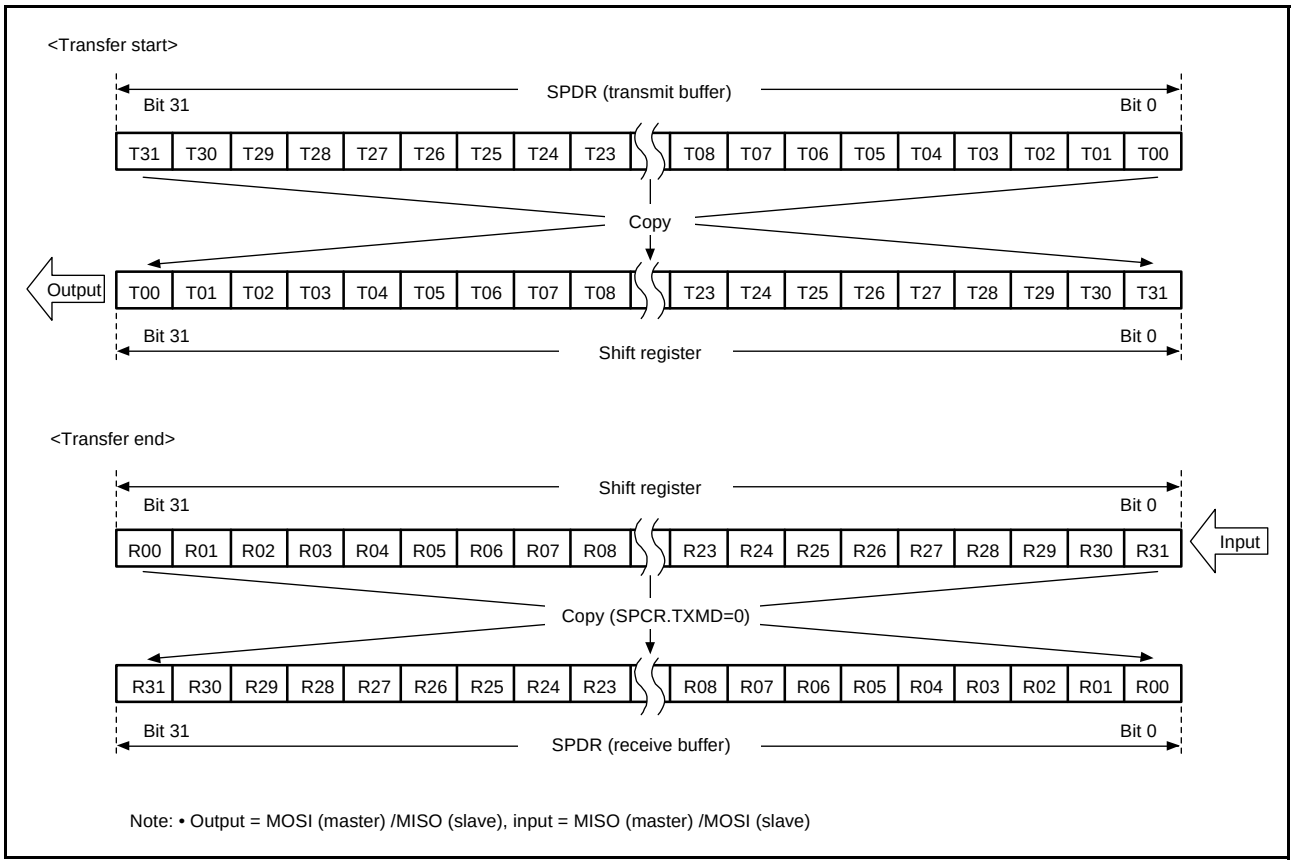


Figure 37.17 LSB-First Transfer (32-Bit Data, Parity Disabled)

(4) LSB-First Transfer (24-Bit Data)

Figure 37.18 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 eight bits of the transmit buffer are stored in the higher-order eight 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 eight bits of the receive buffer.

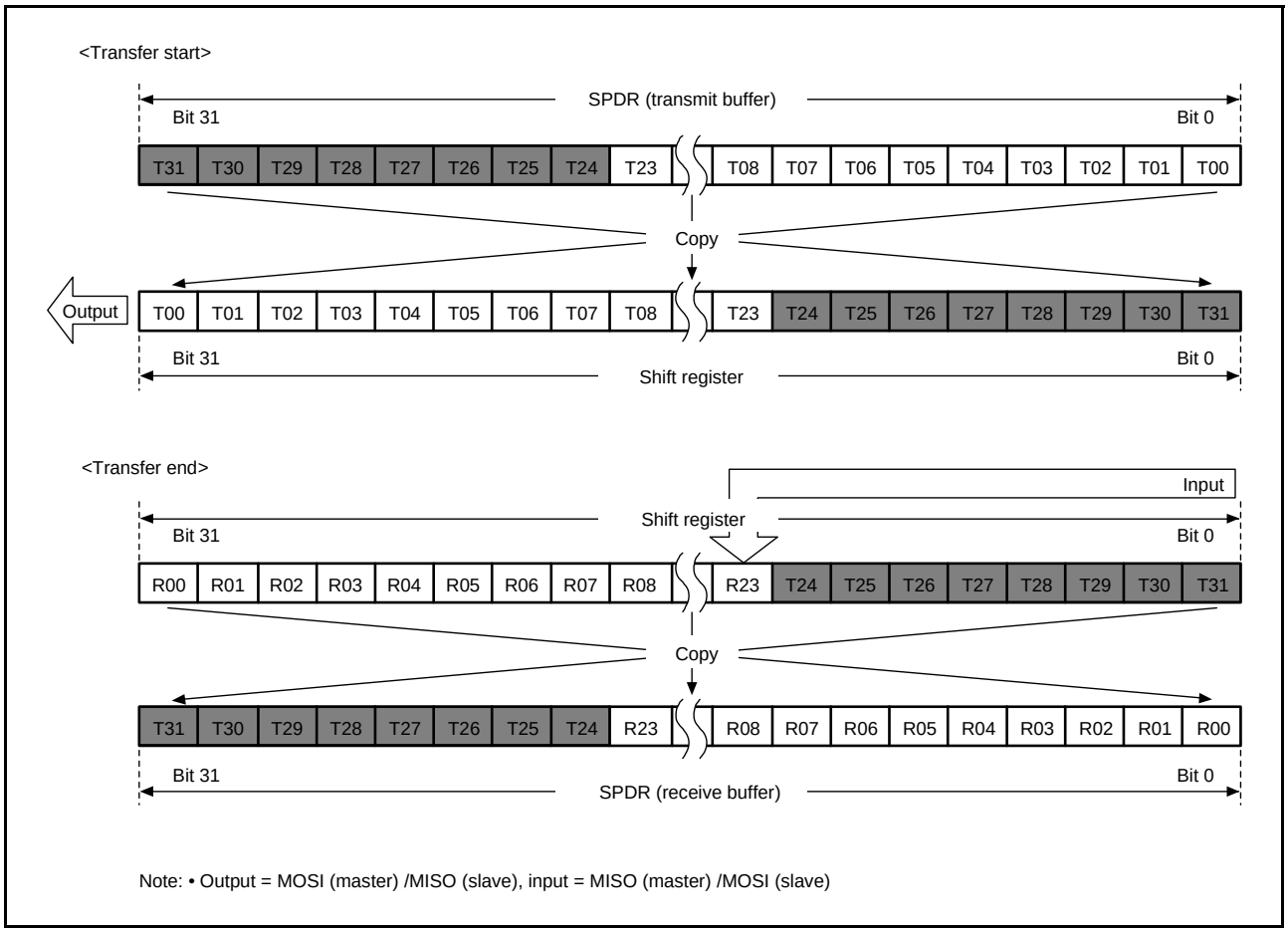


Figure 37.18 LSB-First Transfer (24-Bit Data, Parity Disabled)

37.3.4.2 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 37.19 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.

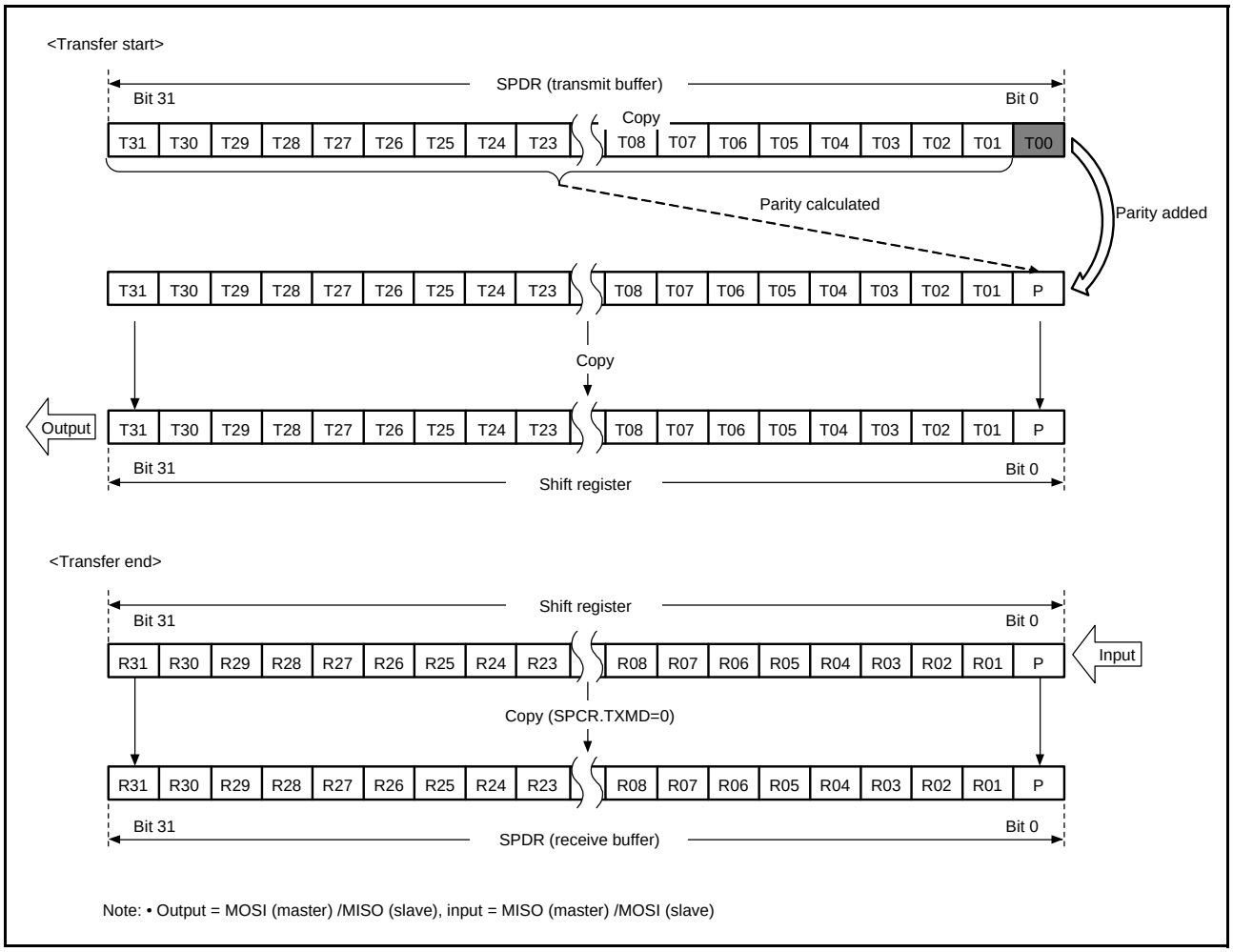


Figure 37.19 MSB-First Transfer (32-Bit Data, Parity Enabled)

(2) MSB-First Transfer (24-Bit Data)

Figure 37.20 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 T24 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 eight bits of the transmit buffer are stored in the higher-order eight 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 eight bits of the receive buffer.

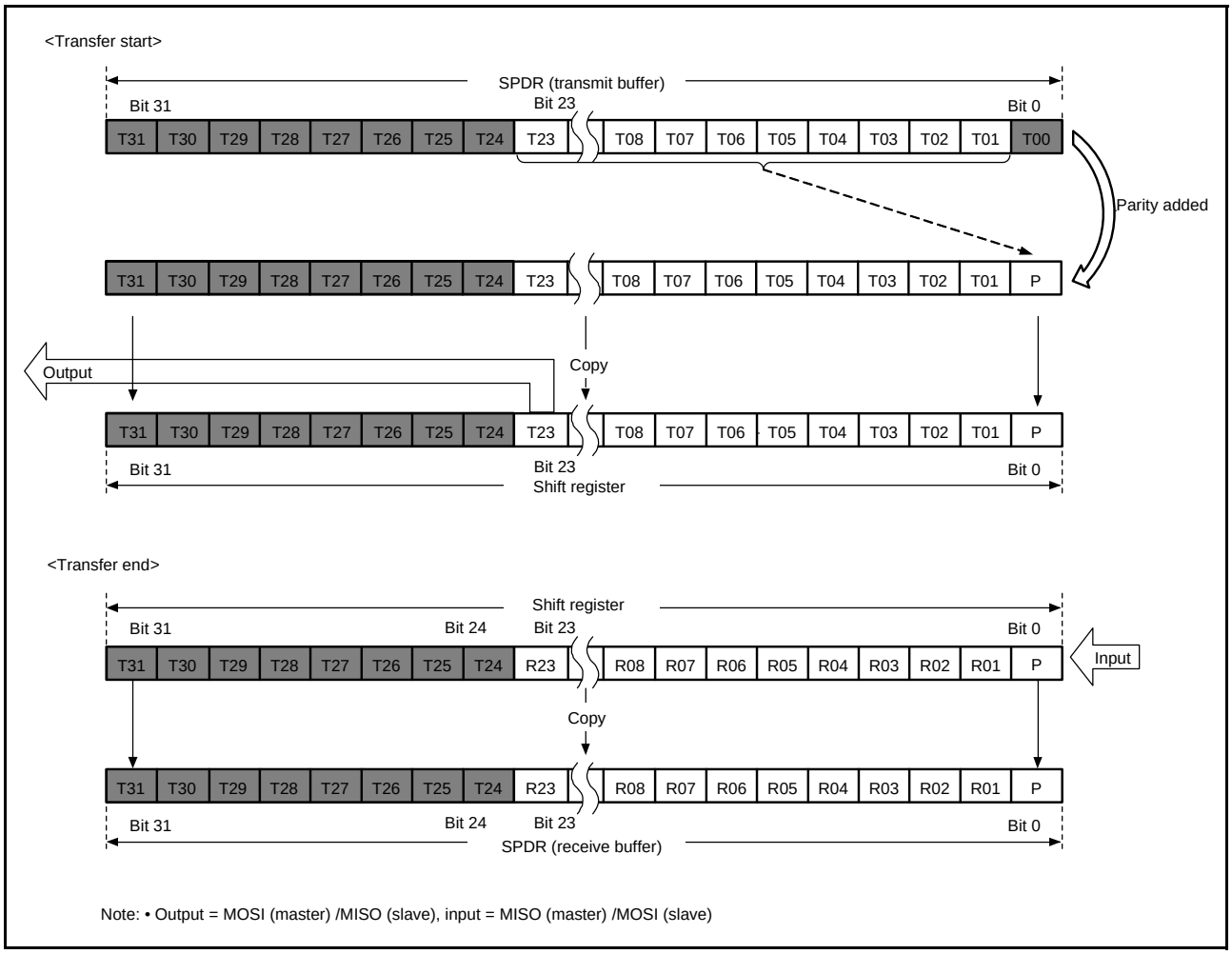


Figure 37.20 MSB-First Transfer (24-Bit Data, Parity Enabled)

(3) LSB-First Transfer (32-Bit Data)

Figure 37.21 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 T01. 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 R31 to P are checked by judging the parity.

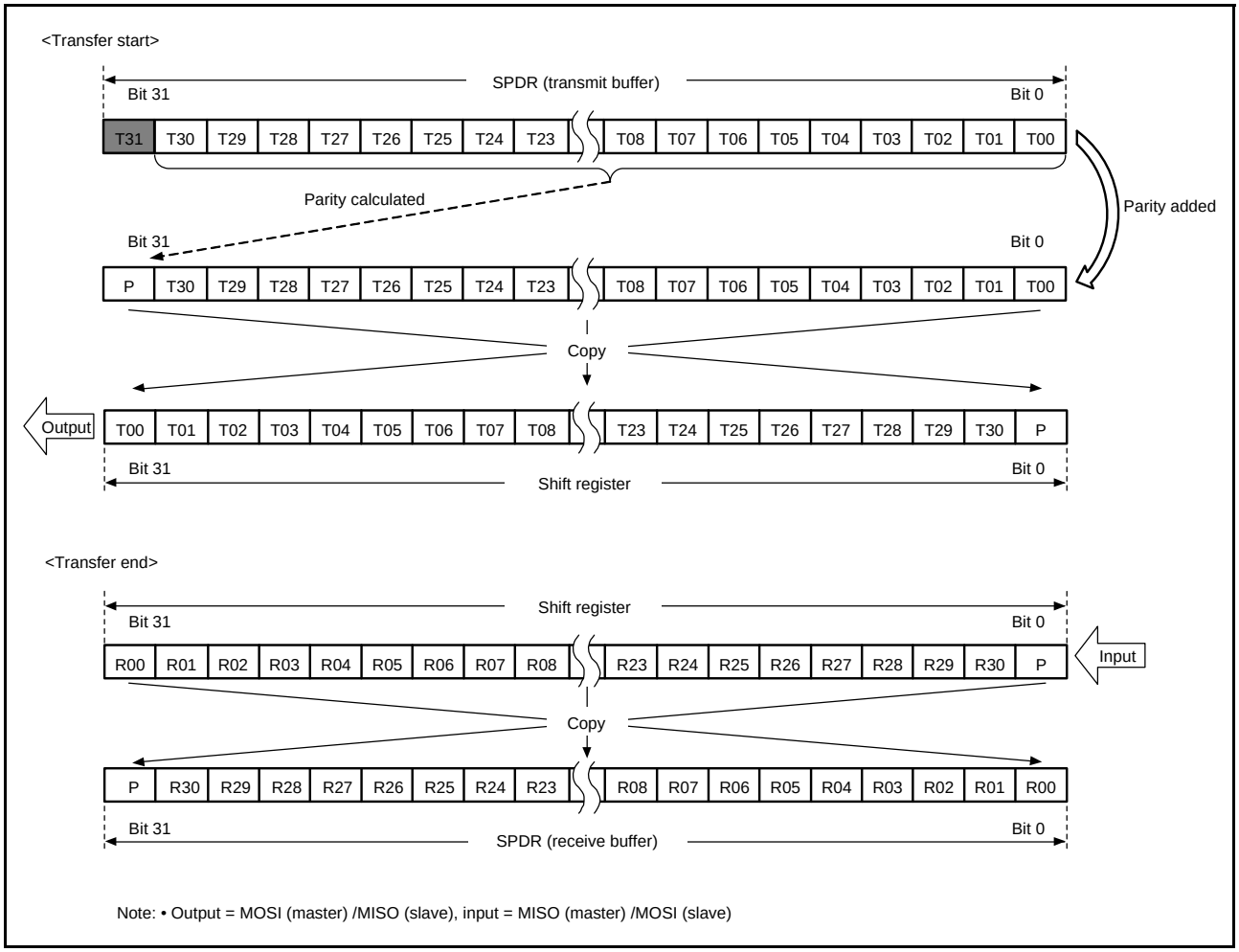


Figure 37.21 LSB-First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 37.22 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 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 R00 to P are checked by judging the parity. At this time, the higher-order eight bits of the transmit buffer are stored in the higher-order eight 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 eight bits of the receive buffer.

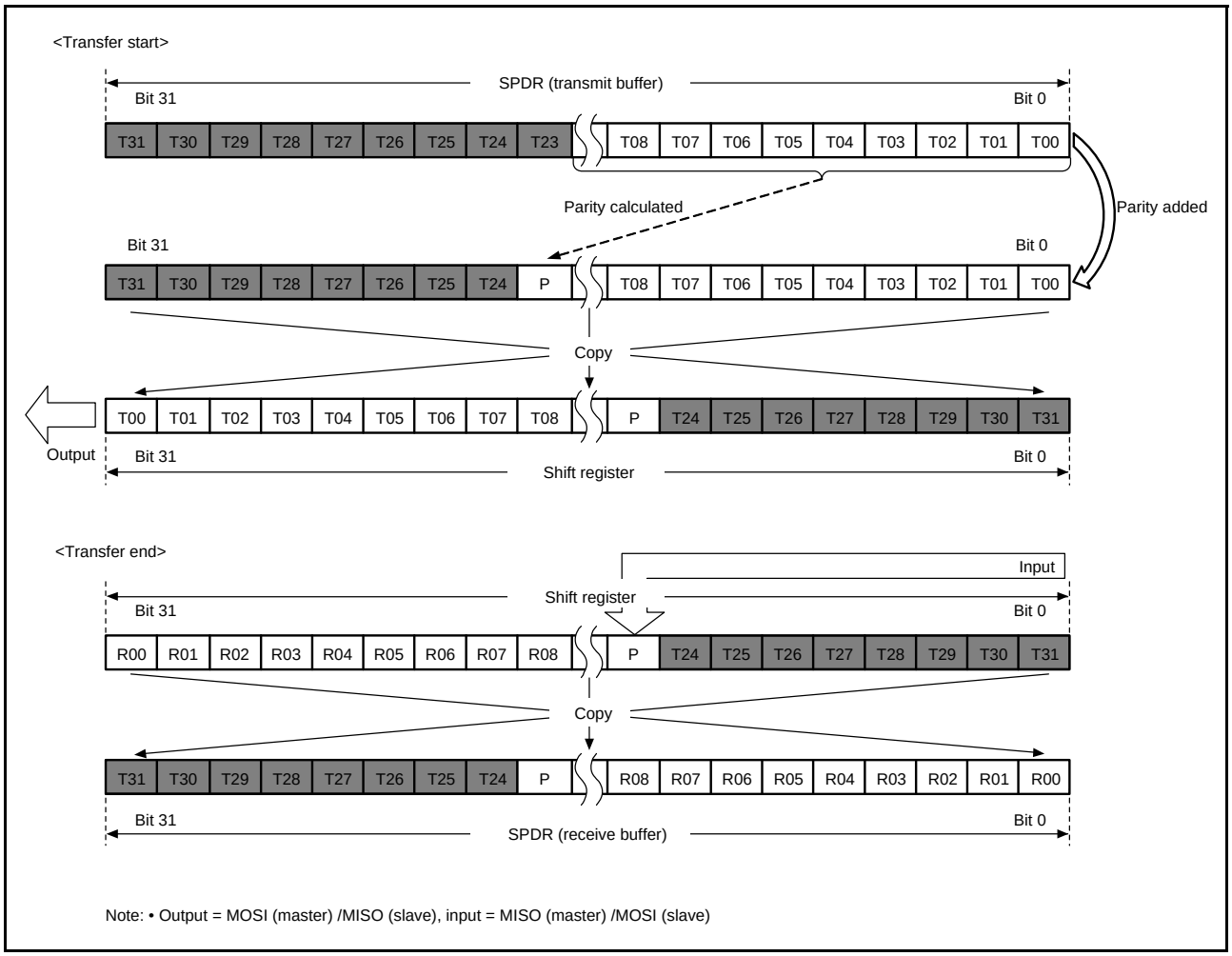


Figure 37.22 LSB-First Transfer (24-Bit Data, Parity Enabled)

37.3.5 Transfer Format

37.3.5.1 CPHA = 0

Figure 37.23 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) is not guaranteed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 37.23, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCKn (CPOL = 1) indicates the RSPCKn 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 input/output directions of the signals depend on the RSPI settings. For details, see section 37.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals commences at an SSLni signal assertion timing. The first RSPCKn signal change timing that occurs after the SSLni signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIn and MISOn signals is always 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLni signal assertion to RSPCKn oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKn oscillation to an SSLni signal negation (SSL negation delay). t3 denotes a period in which SSLni 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 LSI is in master mode, see section 37.3.10.1, Master Mode Operation.

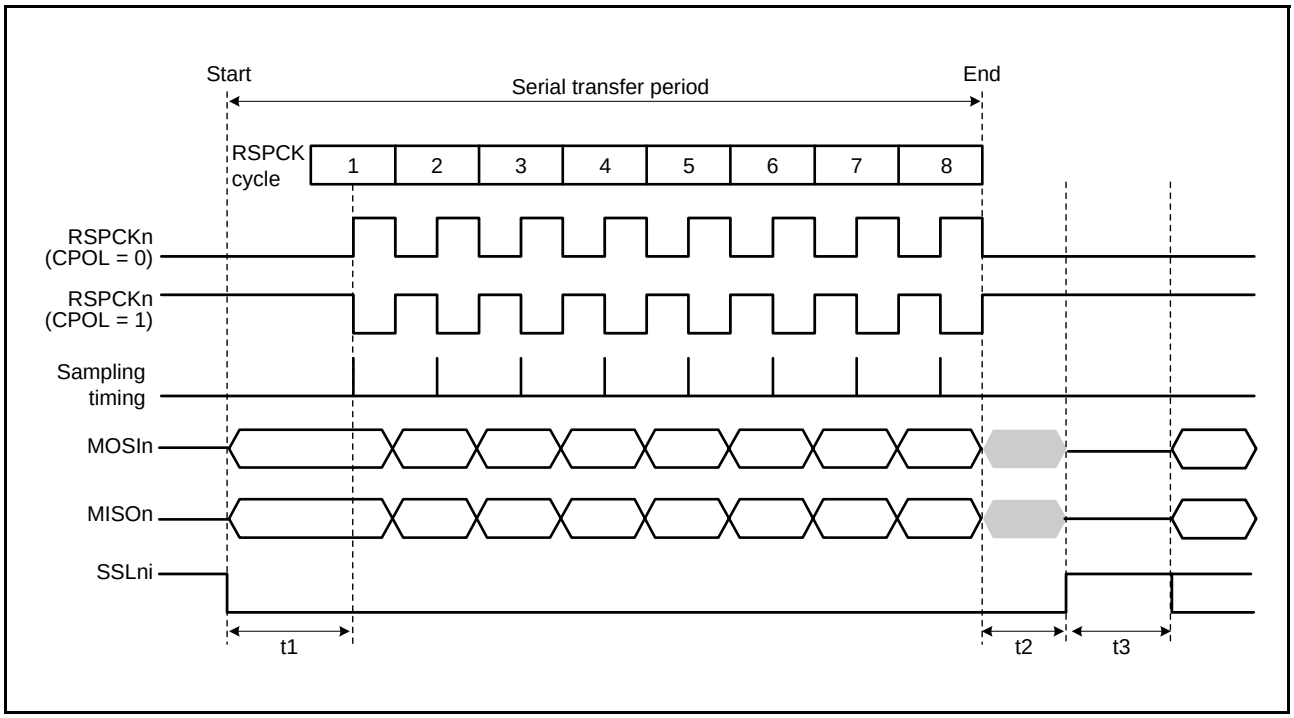


Figure 37.23 RSPI Transfer Format (CPHA = 0)

37.3.5.2 CPHA = 1

Figure 37.24 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 SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 37.24, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCK (CPOL = 1) indicates the RSPCKn 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 input/output directions of the signals depend on the RSPI mode (master or slave). For details, see section 37.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOn signal commences at an SSLni signal assertion timing. The output of valid data to the MOSIn and MISOn signals commences at the first RSPCKn signal change timing that occurs after the SSLni signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn 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 LSI is in master mode, see section 37.3.10.1, Master Mode Operation.

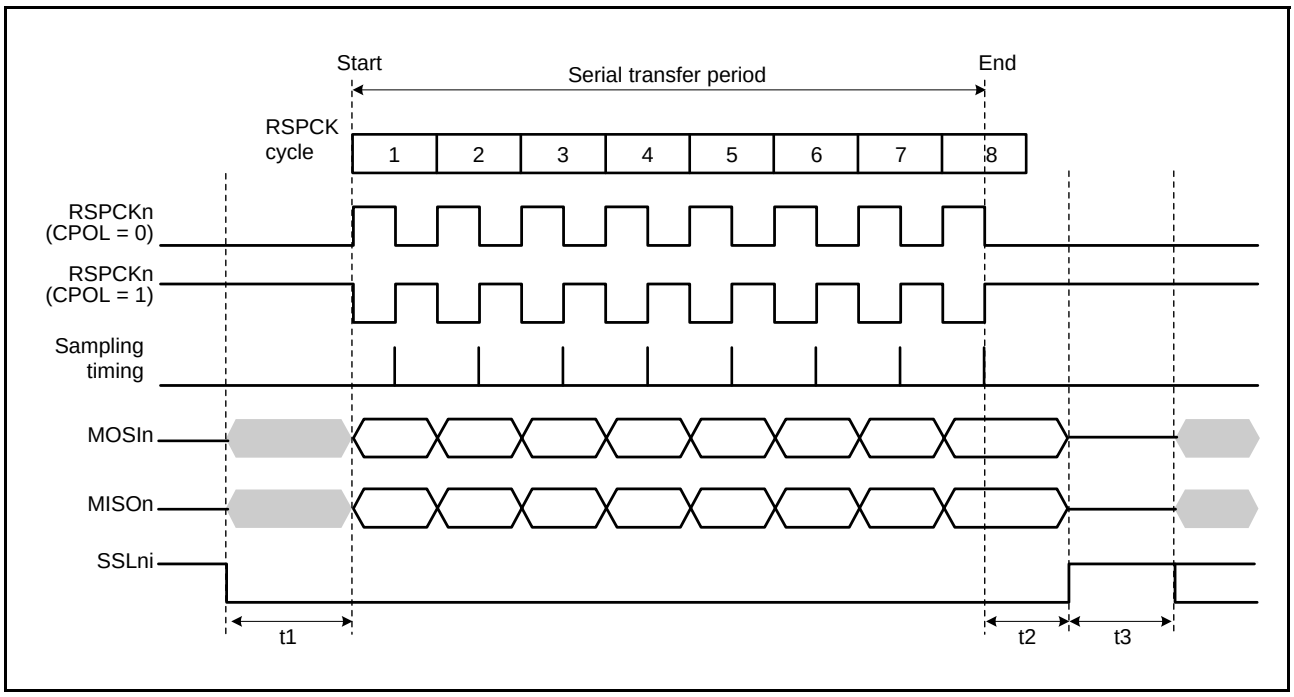


Figure 37.24 RSPI Transfer Format (CPHA = 1)

37.3.6 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 37.25 and Figure 37.26 indicates the condition of access to the SPDR register, where W denotes a write cycle.

37.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 37.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 37.25, 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

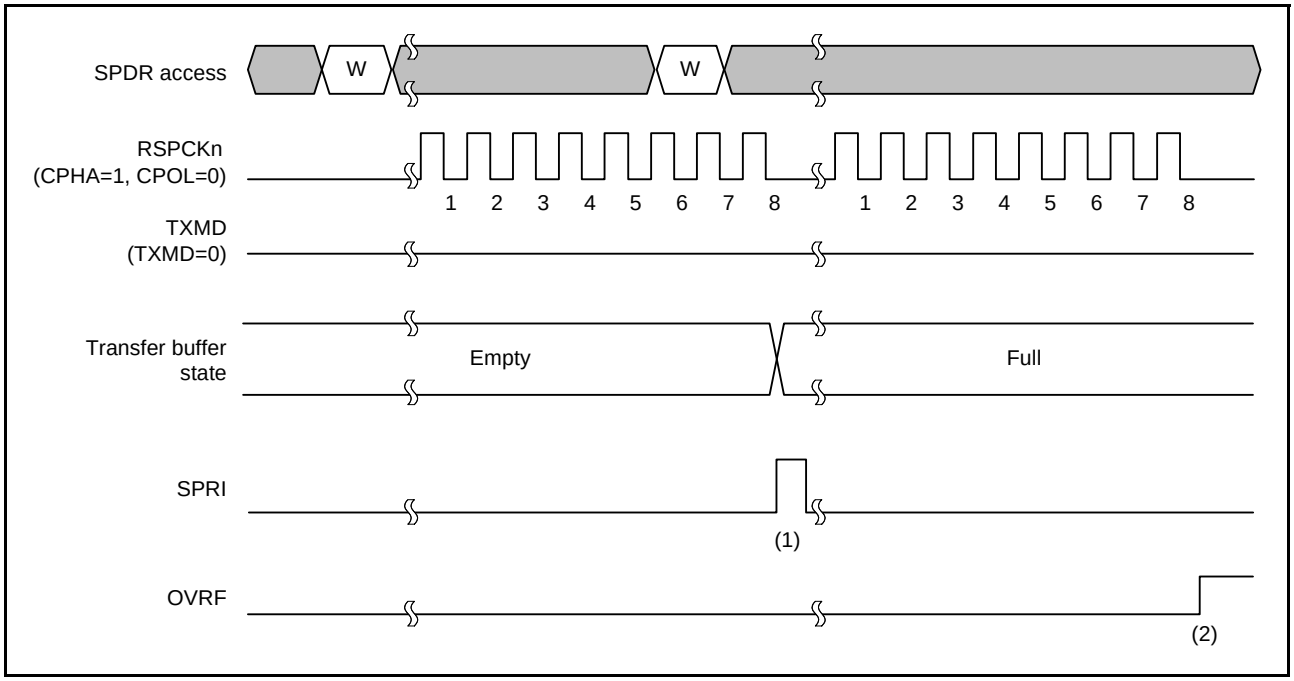


Figure 37.25 Operation Example of SPCR.TXMD = 0

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) 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 performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI transmits transmit data and receives received data. Therefore, the SPSR.OVRF flag is set to 1 at the timings of (1) and (2).

37.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 37.26 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 37.26, 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

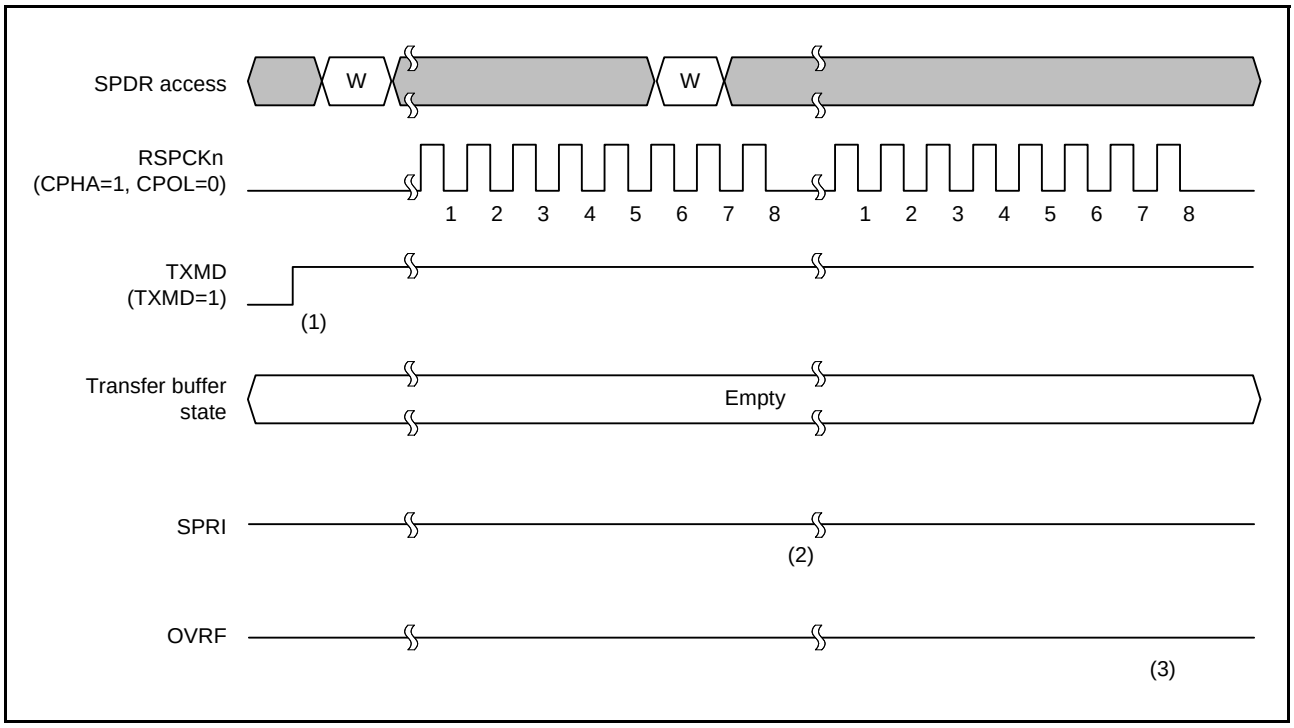


Figure 37.26 Operation Example of SPCR.TXMD = 1

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.OVRF flag is 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 RSPI does not copy the data in 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 transmit data but does not receive received data. Therefore, the SPSR.OVRF flag remains cleared to 0 at the timings of (1) to (3).

37.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 37.27 shows an example of operation of the RSPI transmit buffer empty interrupt (SPTI) and the RSPI receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 37.27 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 37.27, 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

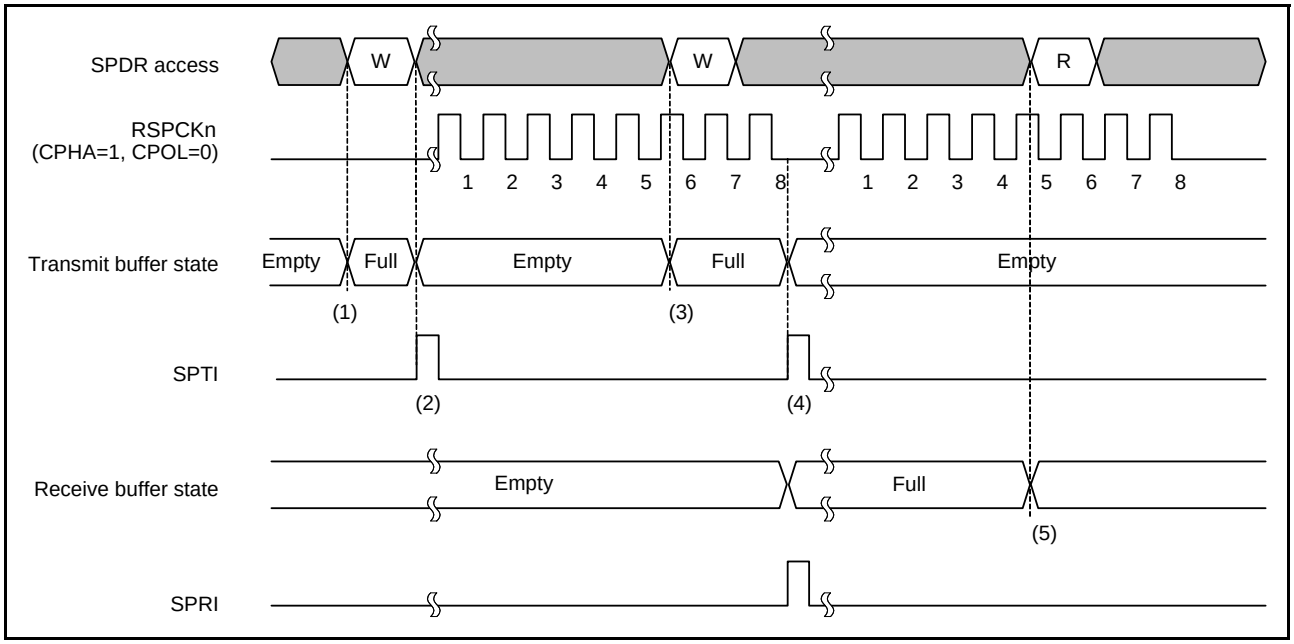


Figure 37.27 Operation Example of SPTI and SPRI Interrupts

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.
2. If the shift register is empty, the RSPI copies the data in the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI). How a serial transfer is started depends on the mode of the RSPI. For details, see section 37.3.10, SPI Operation, and section 37.3.11, Clock Synchronous Operation.
3. When transmit data is written to SPDR by the transmit buffer empty interrupt routine, the data is transferred to the transmit buffer. Because the data being transferred serially is stored in the shift register, the RSPI does not copy the data in 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 in the shift register to the receive buffer and generates a receive buffer full interrupt request (SPRI). 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 copies the data in 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 by the receive buffer full interrupt routine, the receive data can be read.

If SPDR is written to when the transmit buffer holds data that has not yet been transmitted, the RSPI does not update the data in the transmit buffer. When writing to SPDR, make sure to use a transmit buffer empty interrupt request. To use an RSPI transmit interrupt, set the SPTIE bit in SPCR to 1.

If the RSPI function is disabled (the SPE bit in SPCR being 0), set the SPTIE bit to 0.

When serial transfer ends with the receive buffer being full, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (see section 37.3.8, 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 interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the corresponding IRn.IR flags (where n is the vector number) in ICUA can be used to confirm the states of the transmission and reception buffers. See section 15, Interrupt Controller (ICUb), for the vector numbers.

37.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is serially transmitted, and the serially 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 37.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 37.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
A	SPDR is written when the transmit buffer is full.	• The contents of the transmit buffer are kept. • Missing write data.	None
B	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 serially transmitted.	None
C	SPDR is read when the receive buffer is empty.	Previously received serial data is output.	None
D	Serial transfer terminates when the receive buffer is full.	• The contents of the receive buffer are kept. • Missing serial receive data.	Overrun error
E	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
F	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped. • RSPI function is disabled.	Mode fault error
G	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped. • RSPI function is disabled.	Mode fault error
H	The SSLn0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOn output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation A described in Table 37.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, write operations to SPDR should be executed using a transmit interrupt request.

Likewise, the RSPI does not detect an error on operation B. 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 B 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).

Similarly, the RSPI does not detect an error on operation C. To prevent extraneous data from being read, SPDR read operation should be executed using a receive interrupt request.

An overrun error shown in D is described in section 37.3.8.1, Overrun Error. A parity error shown in E is described in section 37.3.8.2, Parity Error. A mode fault error shown in F to H is described in section 37.3.8.3, Mode Fault Error.

For the transmit and receive interrupts, refer to section 37.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

37.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the OVRF flag in SPSR 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 37.28 shows an example of operation of the OVRF flag. The SPSR and SPDR accesses shown in Figure 37.28 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 37.28, 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

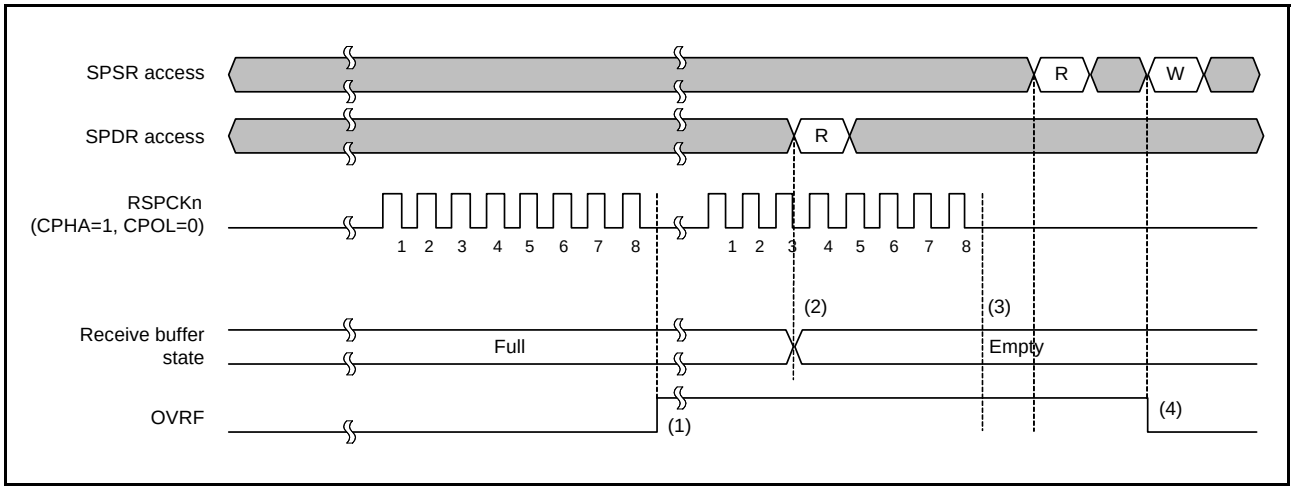


Figure 37.28 Operation Example of OVRF Flag

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 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 can be read. The receive buffer becoming empty does not clear the OVRF flag.
3. If the serial transfer ends with the OVRF flag being 1 (an overrun error), the RSPI does not copy the data in the shift register to the receive buffer. A reception-buffer 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 the value 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, OVRF flag is cleared 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 cleared. The OVRF flag is cleared to 0 under the following condition:

- SPSR is read when the OVRF flag is 1, and then the value 0 is written to the OVRF flag.

37.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit cleared 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 37.29 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 37.29 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 37.29, 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

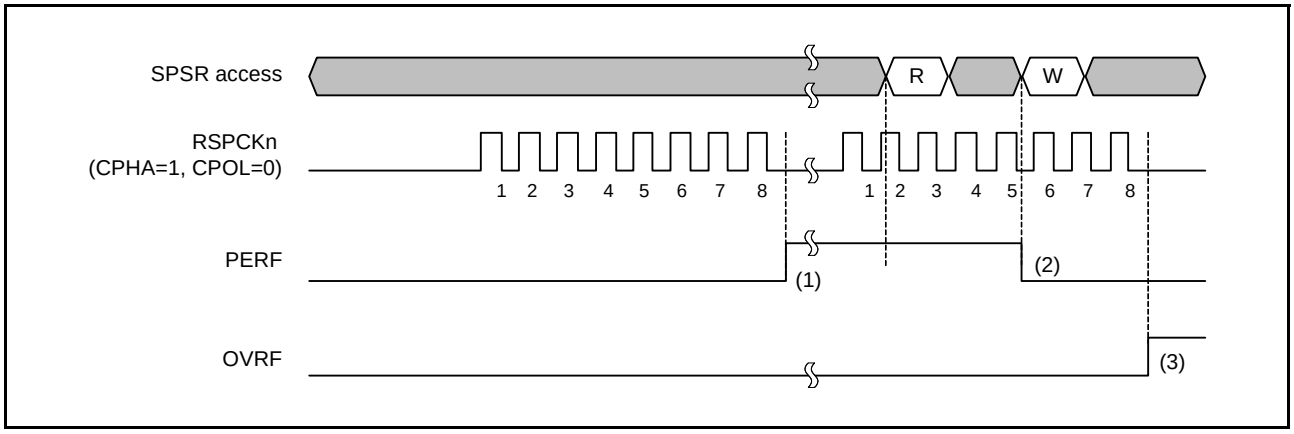


Figure 37.29 Operation Example of PERF Flag

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 the value 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the OVRF flag is cleared 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 SPSR register or by using an RSPI error interrupt and reading 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.
The PERF flag is cleared to 0 under the following condition:

- SPSR register is read when the PERF flag is 1, and then the value 0 is written to the PERF flag.

37.3.8.3 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 SSLn0 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 MODF bit in the RSPI status register (SPSR) 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 SSLn0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP).
When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit in the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLn0 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 (see section 37.3.9, 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. To detect a mode fault error, it is necessary to poll 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 bit is 1, the writing of the value 1 to the SPE bit is ignored. To enable the RSPI function after the detection of a mode fault error, the MODF bit must be set to 0.
The MODF bit is cleared to 0 under the following condition:

- SPSR is read when the MODF flag is 1, and then the value 0 is written to the MODF bit.

37.3.9 Initializing RSPI

If the value 0 is written to the SPE bit in the RSPI control register (SPCR) or the RSPI clears 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 SPE bit in SPCR and initialization by a system reset.

37.3.9.1 Initialization by Clearing the SPE Bit

When the SPE bit in SPCR is cleared, 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

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 OVRF and MODF flags in SPSR 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. Therefore, if the SPTIE bit in SPCR is set to 1 after RSPI initialization, an RSPI transmit interrupt is generated. When the RSPI is initialized by the CPU, in order to disable any RSPI transmit interrupt, the value 0 should be written to the SPTIE bit simultaneously with the writing of the value 0 to the SPE bit. To disable any RSPI transmit interrupt after a mode fault error is detected, use an error handling routine to write the value 0 to the SPTIE bit.

37.3.9.2 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 37.3.9.1, Initialization by Clearing the SPE Bit.

37.3.10 SPI Operation

37.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (see section 37.3.8, 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 (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 transmission 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, see section 37.3.5, Transfer Format. The polarity of the SSLni output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

Irrespective of the CPHA bit in the RSPI command register (SPCMDm), the RSPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX), 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. The polarity of the SSLni output pin depends on the SSLP register settings.
For details on the RSPI transfer format, see section 37.3.5, 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: SSLni 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.

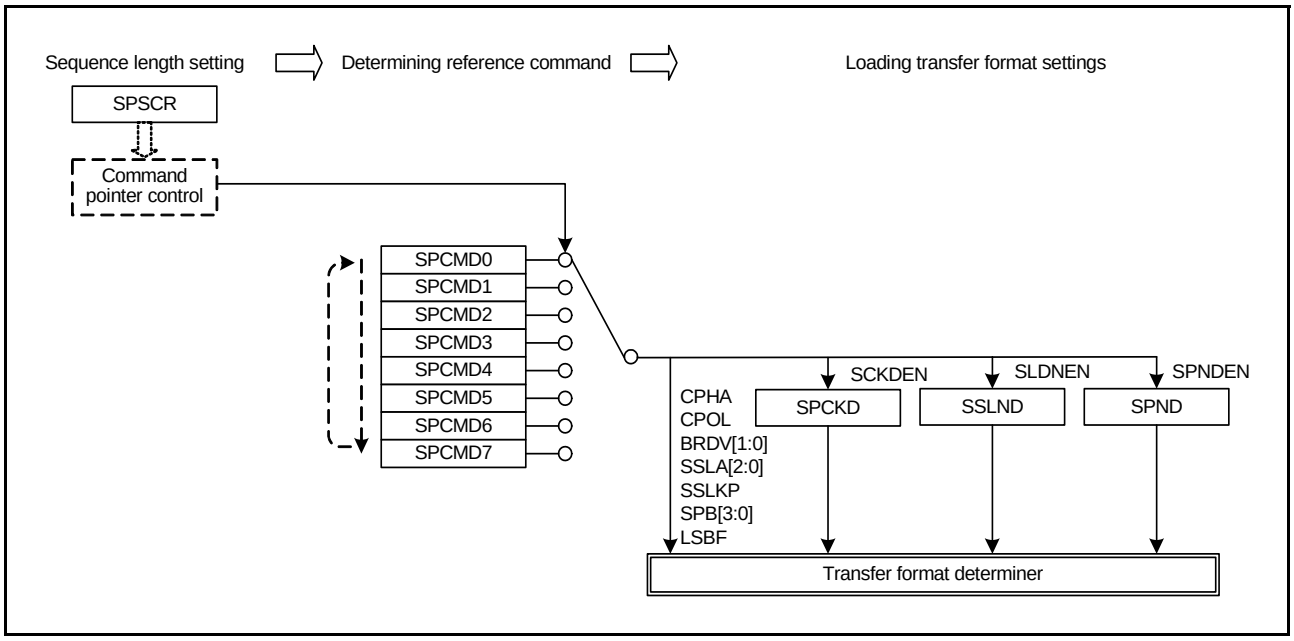


Figure 37.30 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

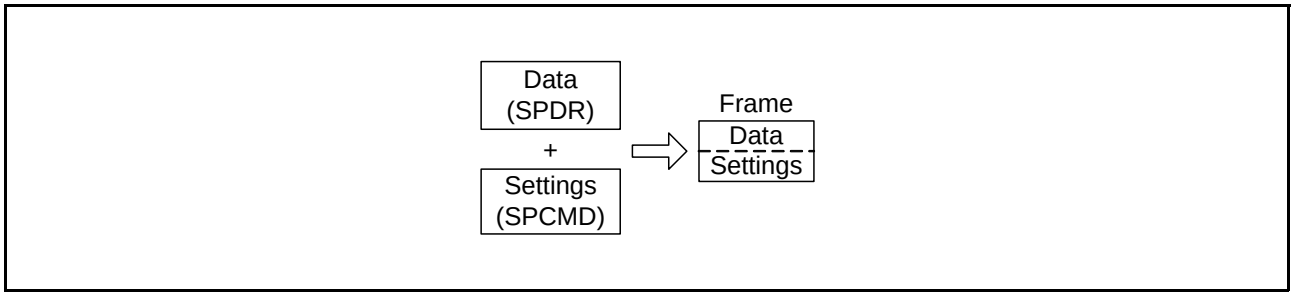


Figure 37.31 Concept of a Frame

Figure 37.32 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 37.4.

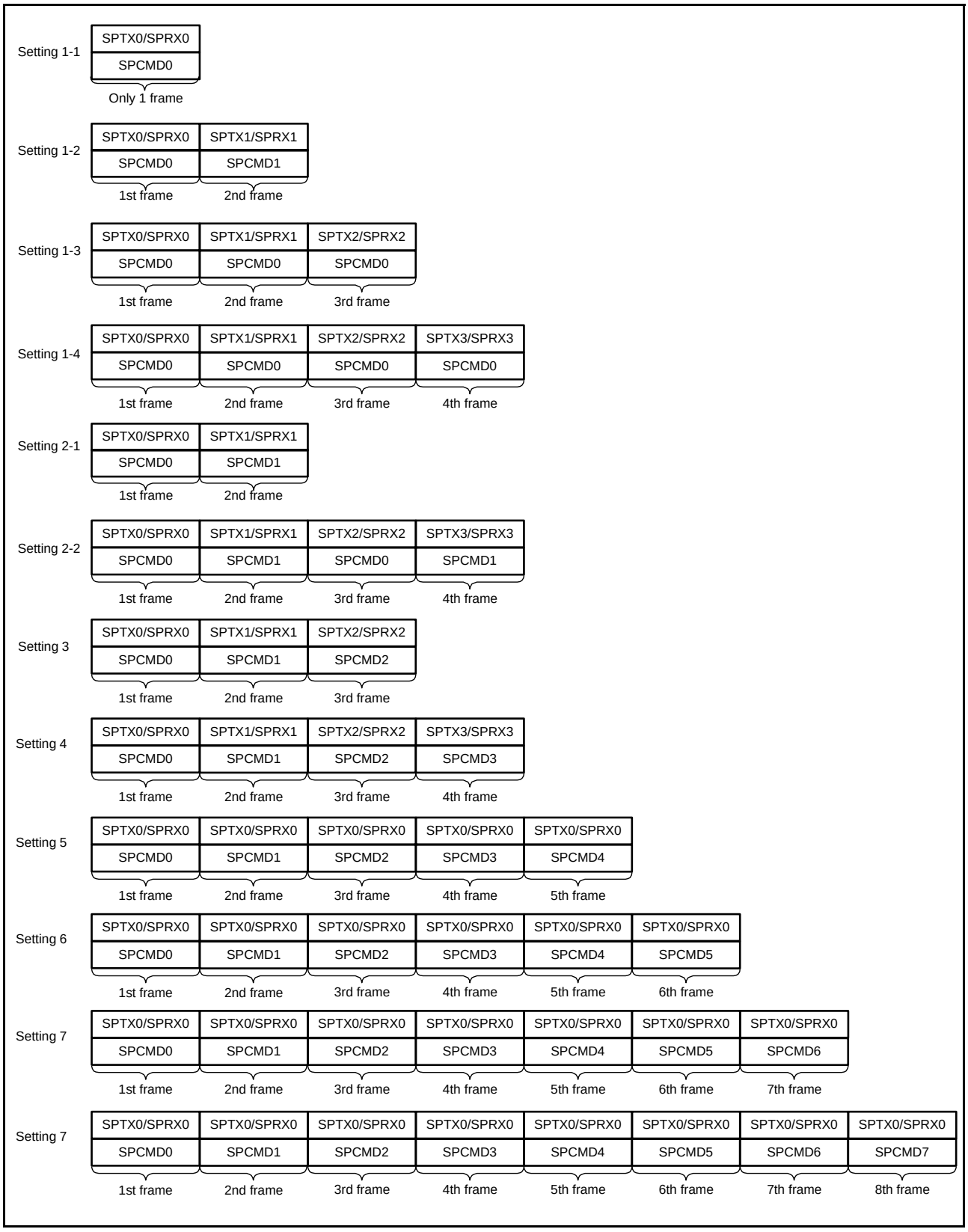


Figure 37.32 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

If the SSLKP bit in the RSPI command register (SPCMDm) that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

Figure 37.33 shows an example of an SSLni 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 37.33. It should be noted that the polarity of the SSLni output signal depends on the SSLP register settings.

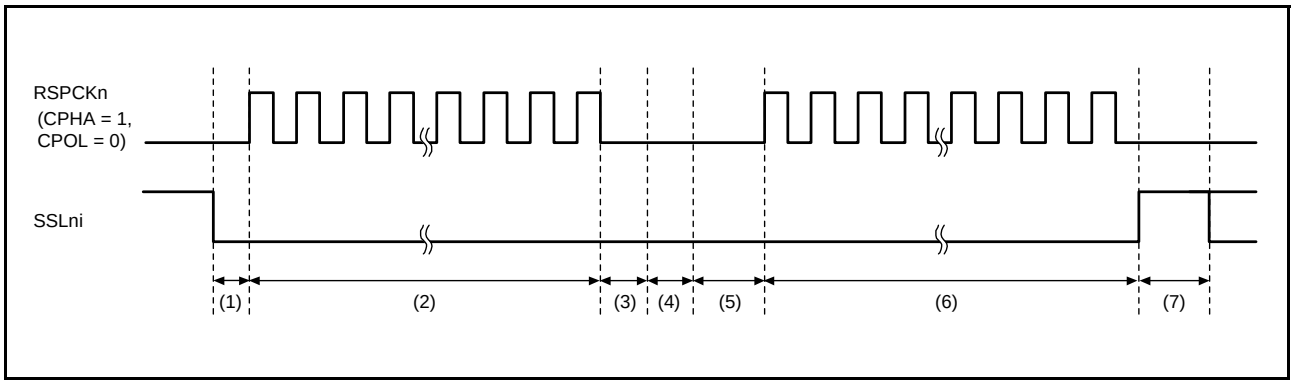


Figure 37.33 Example of Burst Transfer Operation using SSLKP Bit

(1) Based on SPCMD0, the RSPI asserts the SSLni 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 SSLni 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 SSLni 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 SSLni signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLni signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLni signal status to SSLni signal assertion ((5) in Figure 37.33) corresponding to the command for the next transfer. Note that if such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels may occur.

The RSPI in master mode references the SSLni 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 SSLni signal assertion for the next transfer that is detected internally. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 37.3.10, SPI Operation).

(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 37.9. For a definition of RSPCK delay, see section 37.3.5, Transfer Format.

Table 37.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	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 37.10. For a definition of SSL negation delay, see section 37.3.5, Transfer Format.

Table 37.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	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 37.11. For a definition of next-access delay, see section 37.3.5, Transfer Format.

Table 37.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000 to 111	1 RSPCK + 2 PCLK
1	000	1 RSPCK + 2 PCLK
	001	2 RSPCK + 2 PCLK
	010	3 RSPCK + 2 PCLK
	011	4 RSPCK + 2 PCLK
	100	5 RSPCK + 2 PCLK
	101	6 RSPCK + 2 PCLK
	110	7 RSPCK + 2 PCLK
	111	8 RSPCK + 2 PCLK

(8) Initialization Flowchart

Figure 37.34 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, see the descriptions given in the individual blocks.

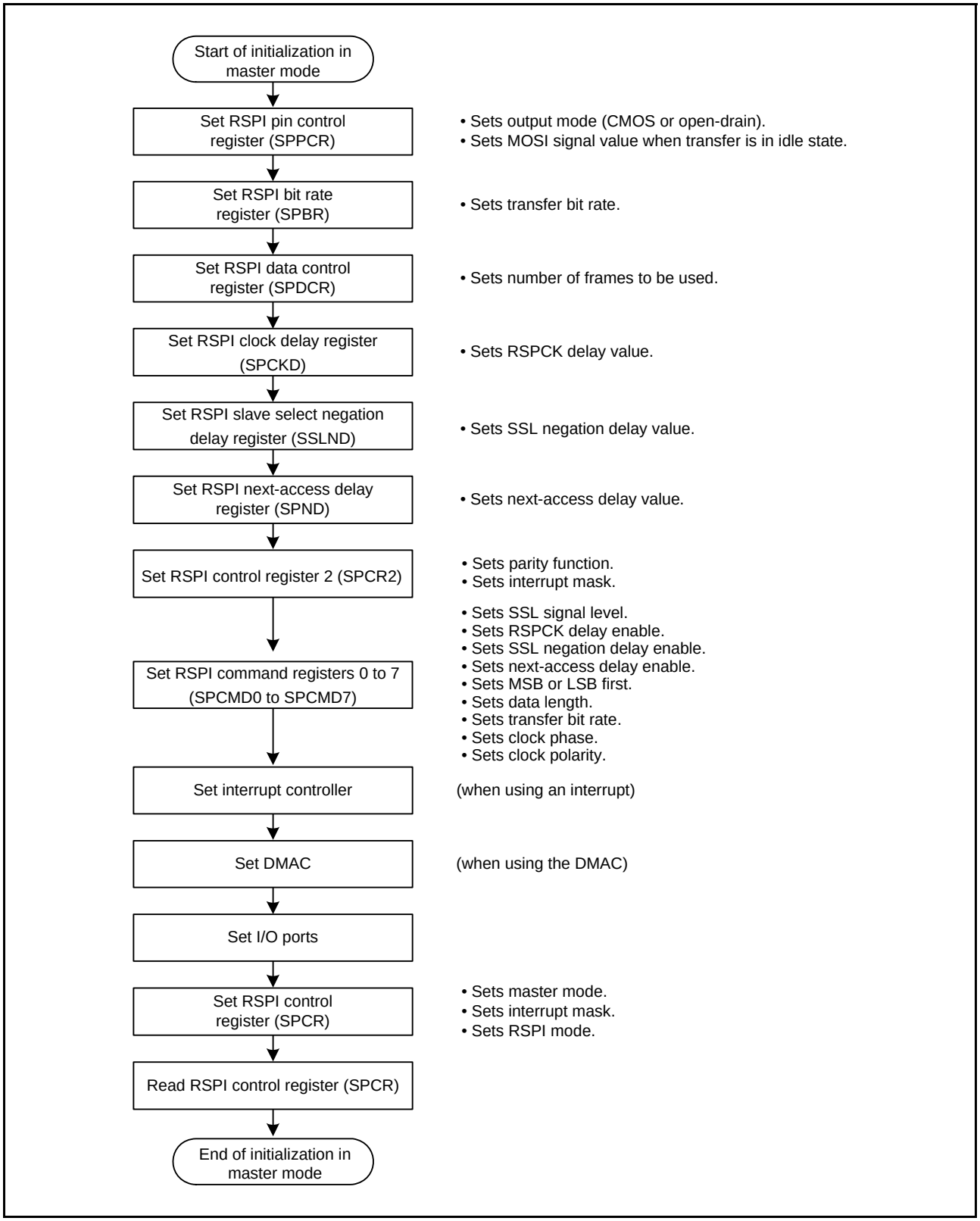
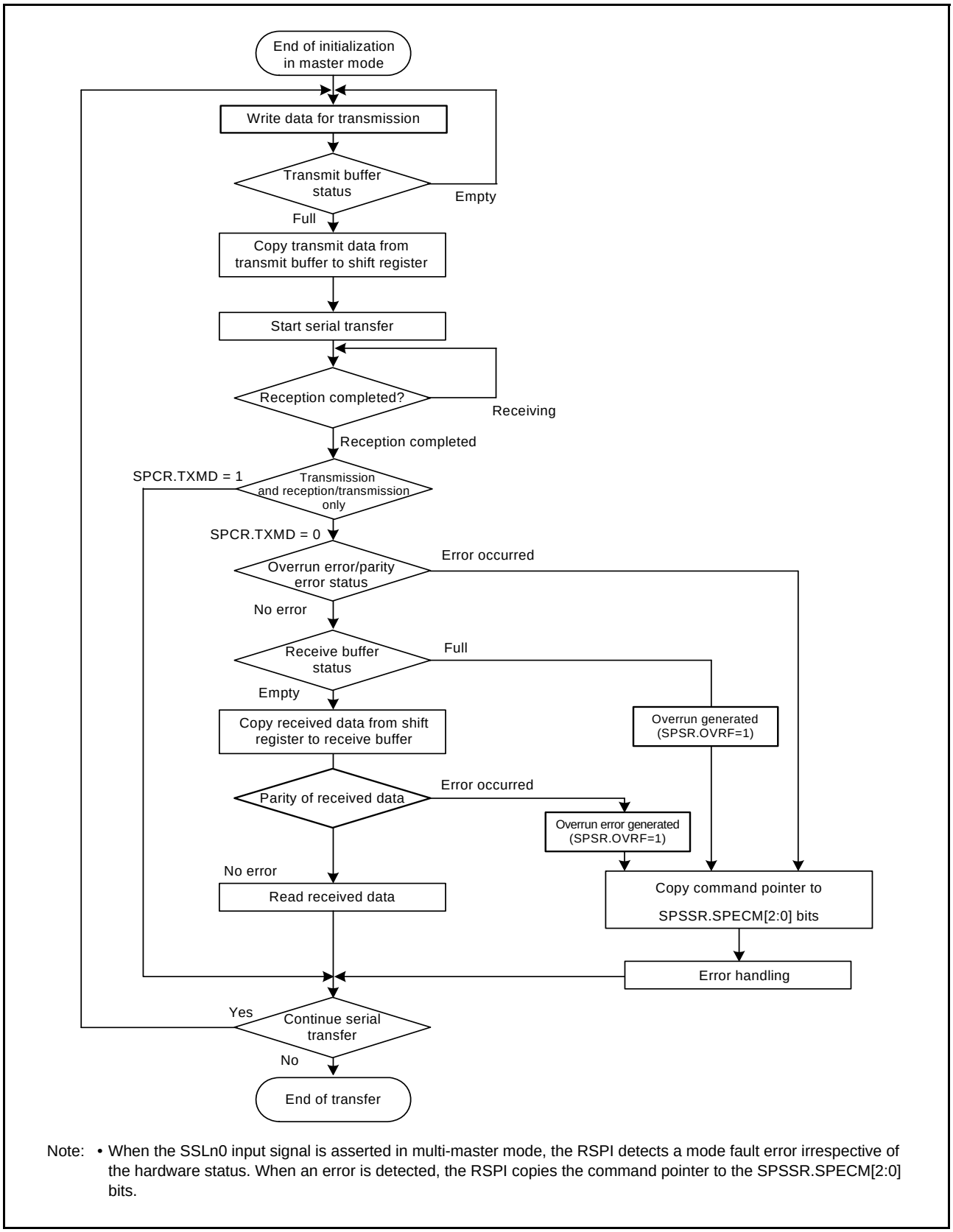


Figure 37.34 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Flowchart of Operations

Figure 37.35 is a flowchart illustrating SPI operations in master mode.



Note: • When the SSLn0 input signal is asserted in multi-master mode, the RSPI detects a mode fault error irrespective of the hardware status. When an error is detected, the RSPI copies the command pointer to the SPSSR.SPECM[2:0] bits.

Figure 37.35 Example of Transfer Operation Flowchart in Master Mode (SPI Operation)

37.3.10.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLn0 input signal assertion, the RSPI needs to start driving valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCKn edge in an SSLn0 signal asserted condition, the RSPI needs to start driving valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 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 CPHA bit setting, the timing at which the RSPI starts driving of the MISOn output signal is the SSLn0 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, see section 37.3.5, Transfer Format. The polarity of the SSLn0 input signal depends on the setting of the SSL0P bit in the RSPI slave select polarity register (SSLP).

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR).
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 SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 37.3.8, 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 SSLn0 input signal depends on the SSLP.SSL0P bit setting.
For details on the RSPI transfer format, see section 37.3.5, 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 SSLn0 input signal. In the type of configuration shown in Figure 37.7 as an example, if the RSPI is used in single-slave mode, the SSLn0 signal is always 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 SSLn0 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 SSLn0 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 SSLn0 input signal. If the CPHA bit is 1, the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state corresponds to a serial transfer period. Even when the SSLn0 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 37.36 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, see the descriptions given in the individual blocks.

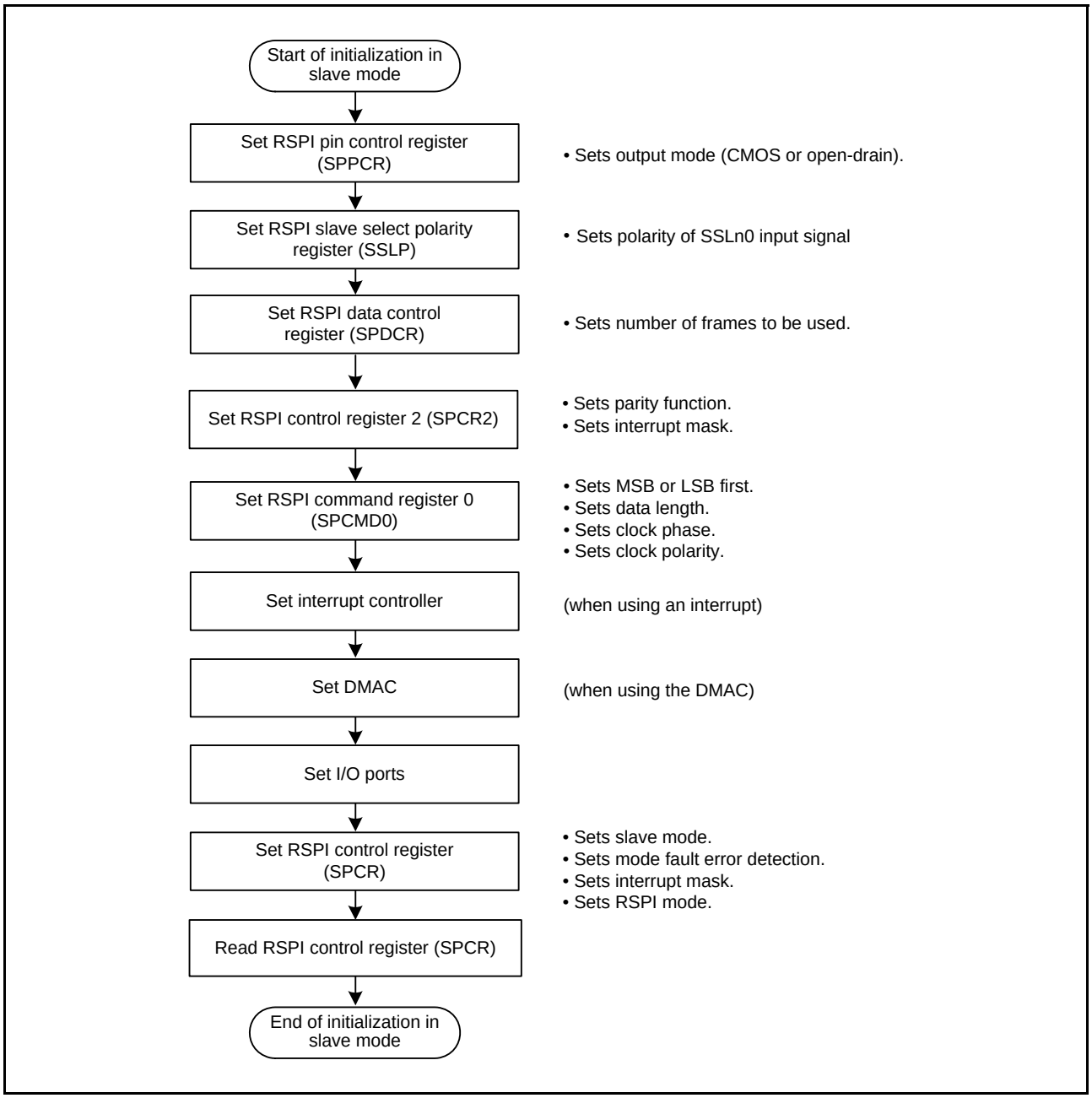


Figure 37.36 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Flowchart of Operations

Figure 37.37 is a flowchart illustrating SPI operations in slave mode.

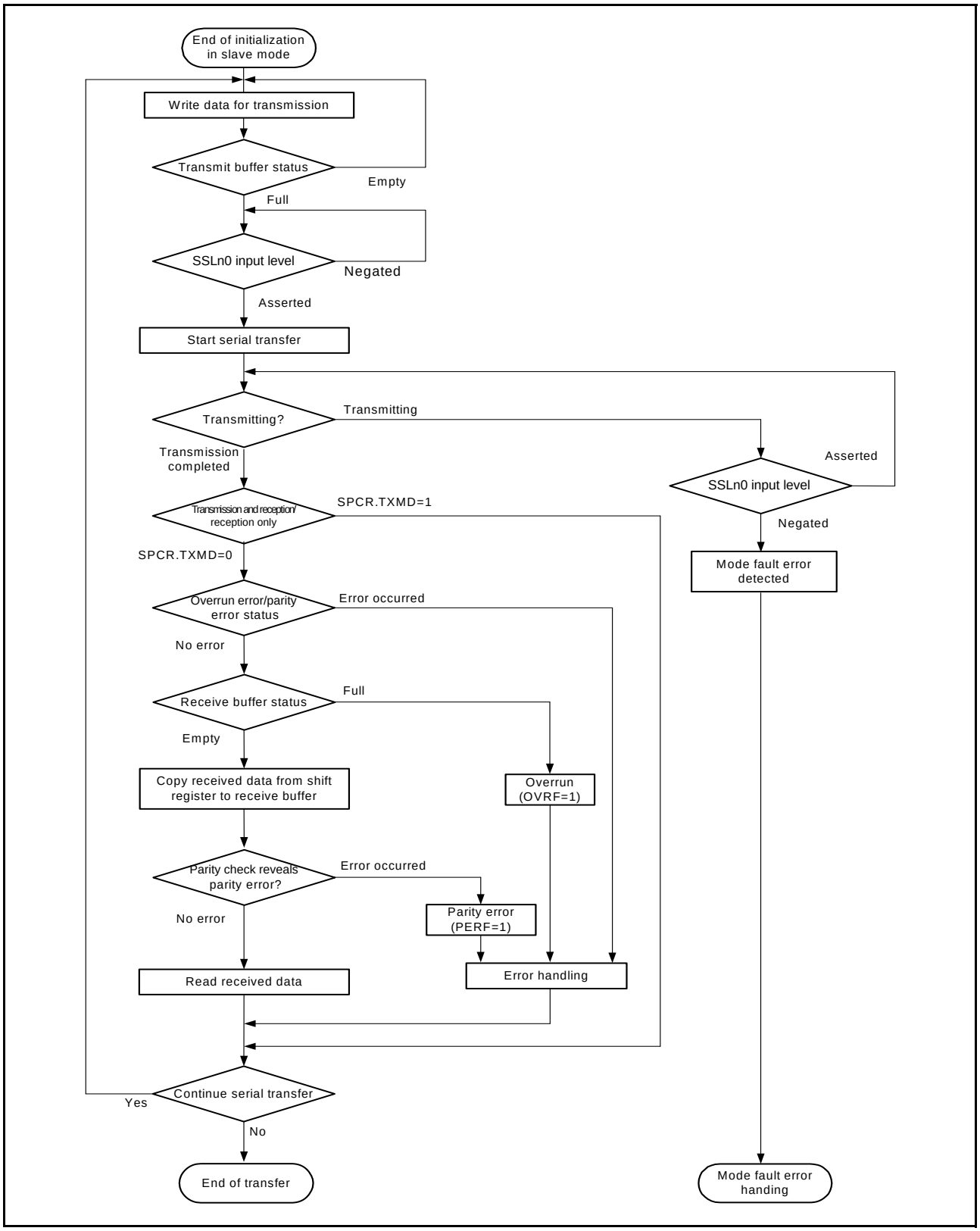


Figure 37.37 Example Flowchart for Transfer Operations in Slave Mode (SPI Operation)

37.3.11 Clock Synchronous Operation

Setting the SPMS bit in the RSPI control register (SPCR) to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLni pin is not used, and the three pins of RSPCKn, MOSIn, and MISOn handle communications. The SSLni pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLni 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 SSLni pin is not used.
Furthermore, operation is not guaranteed if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

37.3.12 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 RSPI data register (SPDR) with the transmit buffer being empty (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 transmission 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, see section 37.3.5, Transfer Format.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If free space is available in the receive buffer, 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, see section 37.3.5, Transfer Format.
However, transfer in clock-synchronous operation is conducted without the SSLn0 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 SSLni 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: SSLni output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKn 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 SPE bit in the RSPI control register (SPCR) 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.

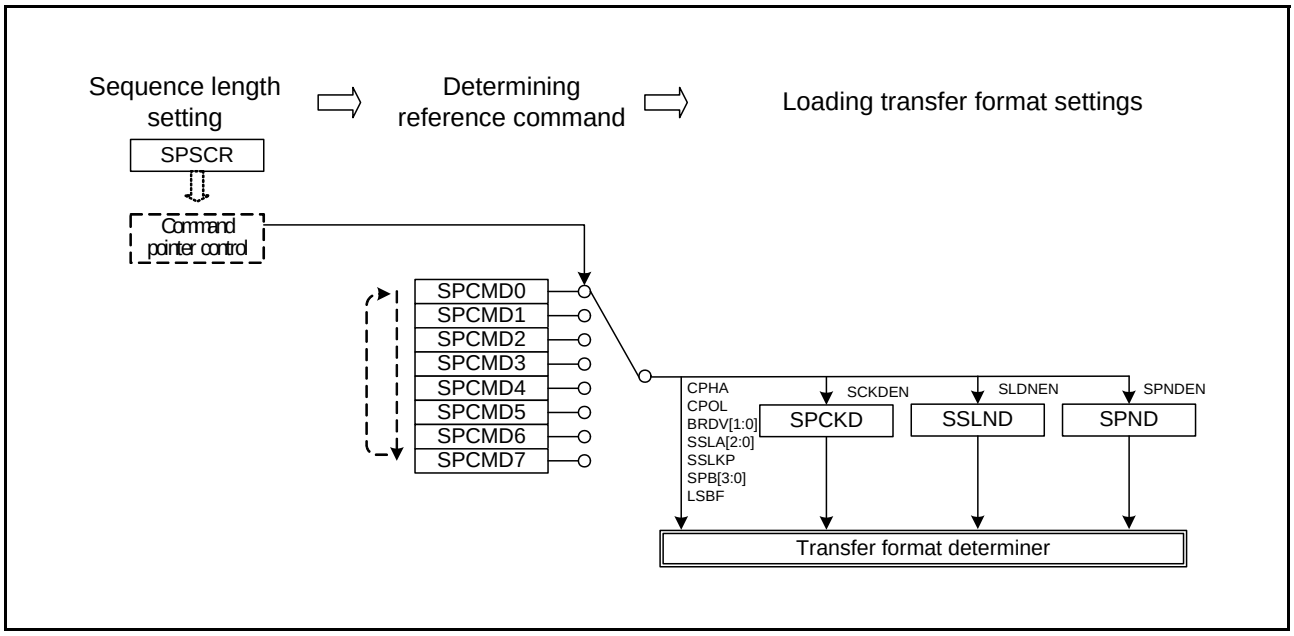


Figure 37.38 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

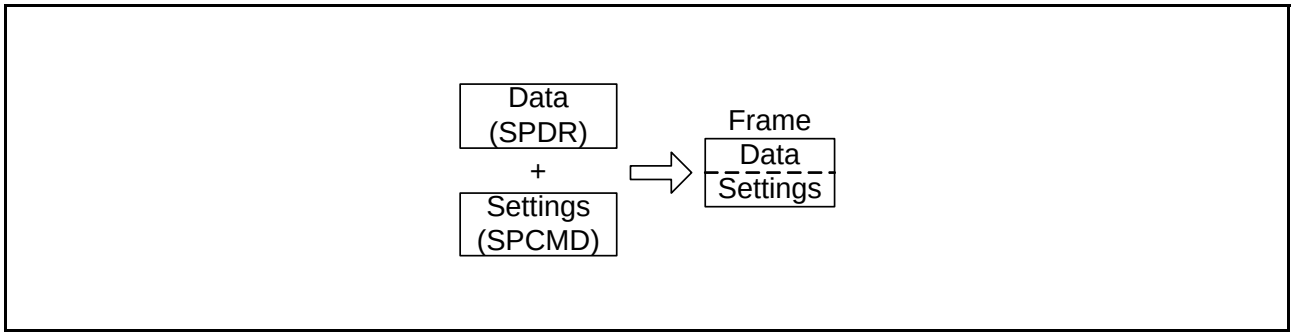


Figure 37.39 Concept of a Frame

Figure 37.40 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 37.4.

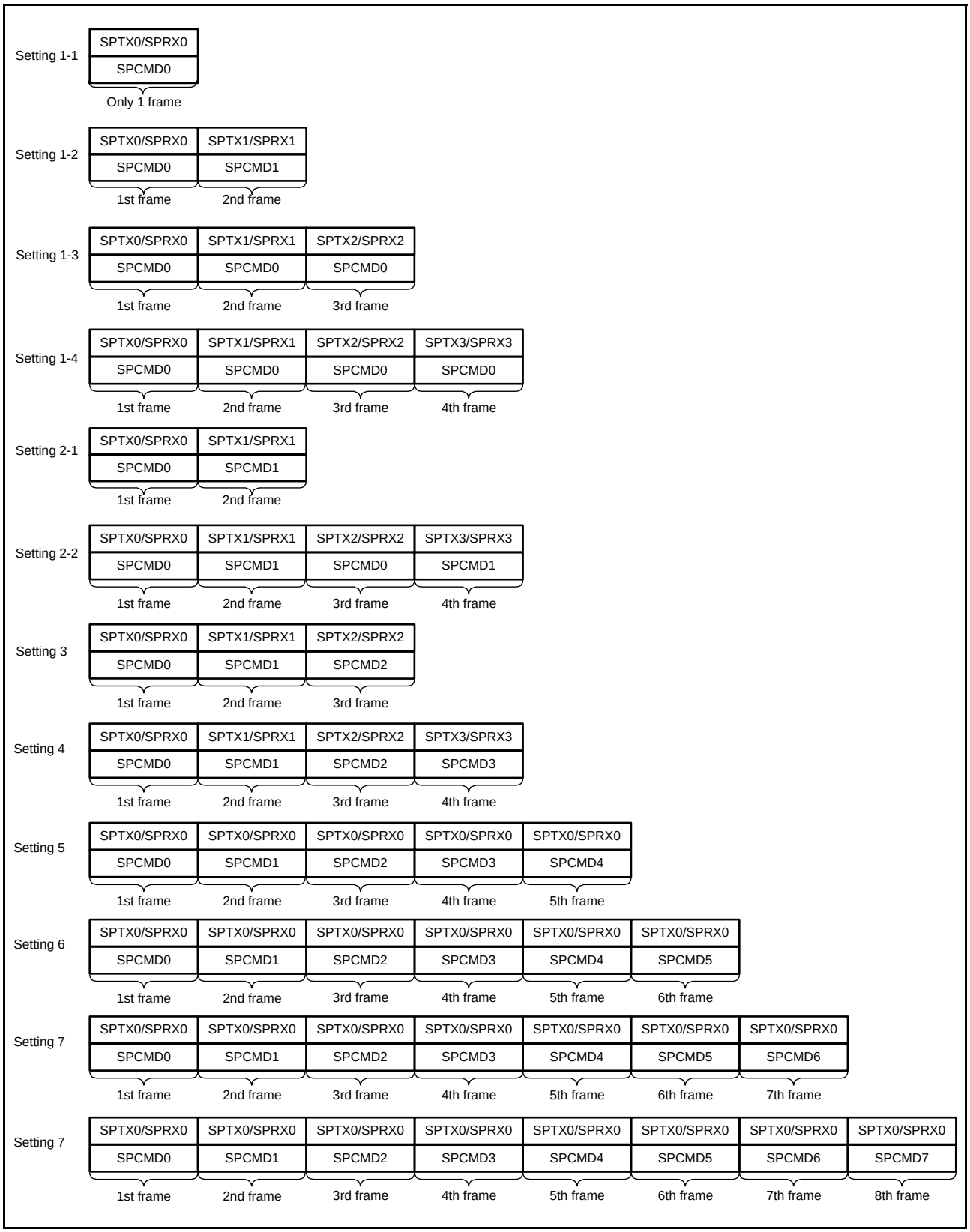


Figure 37.40 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 37.41 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, see the descriptions given in the individual blocks.

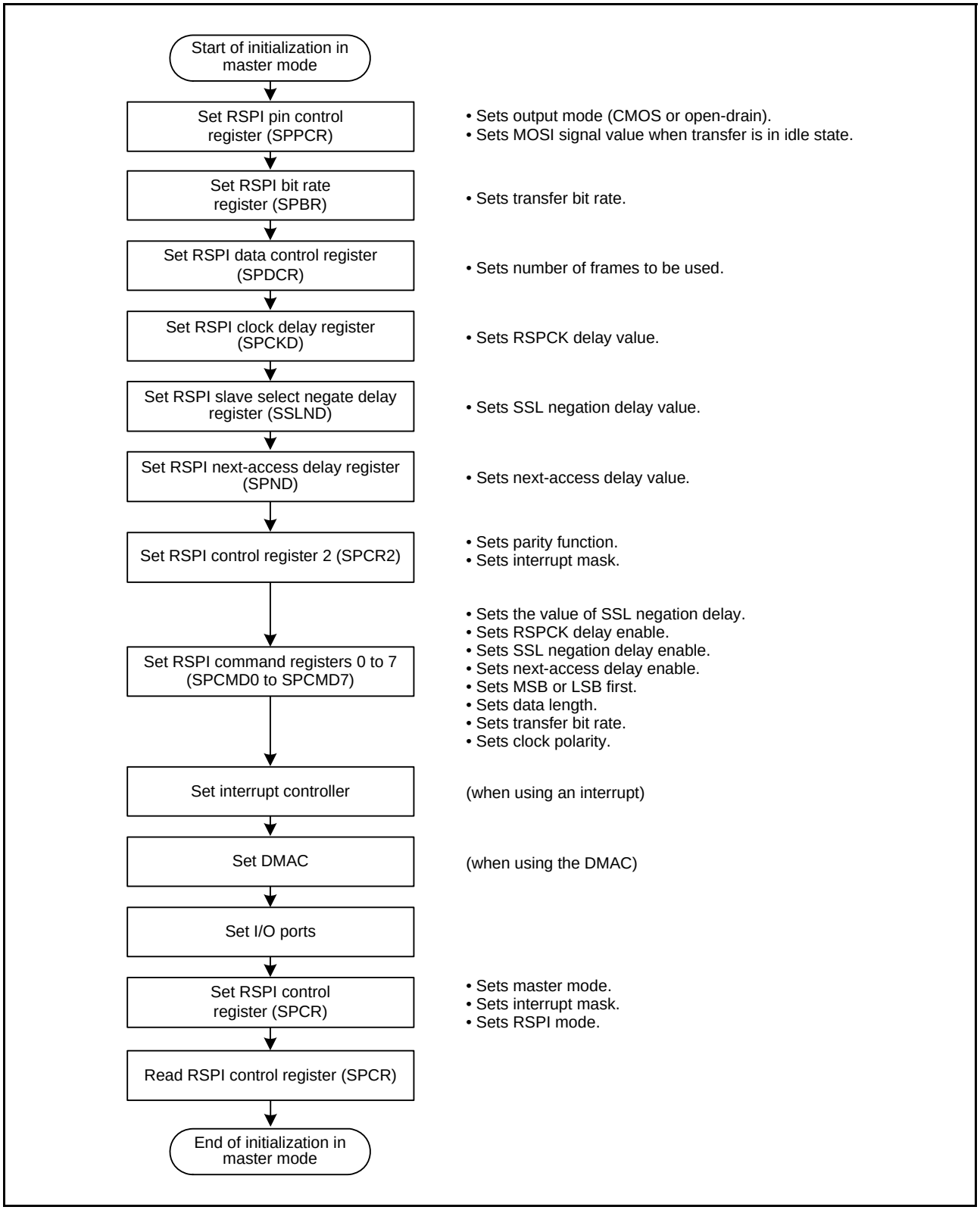


Figure 37.41 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flowchart of Operations

Figure 37.42 is a flowchart illustrating clock synchronous operations in master mode.

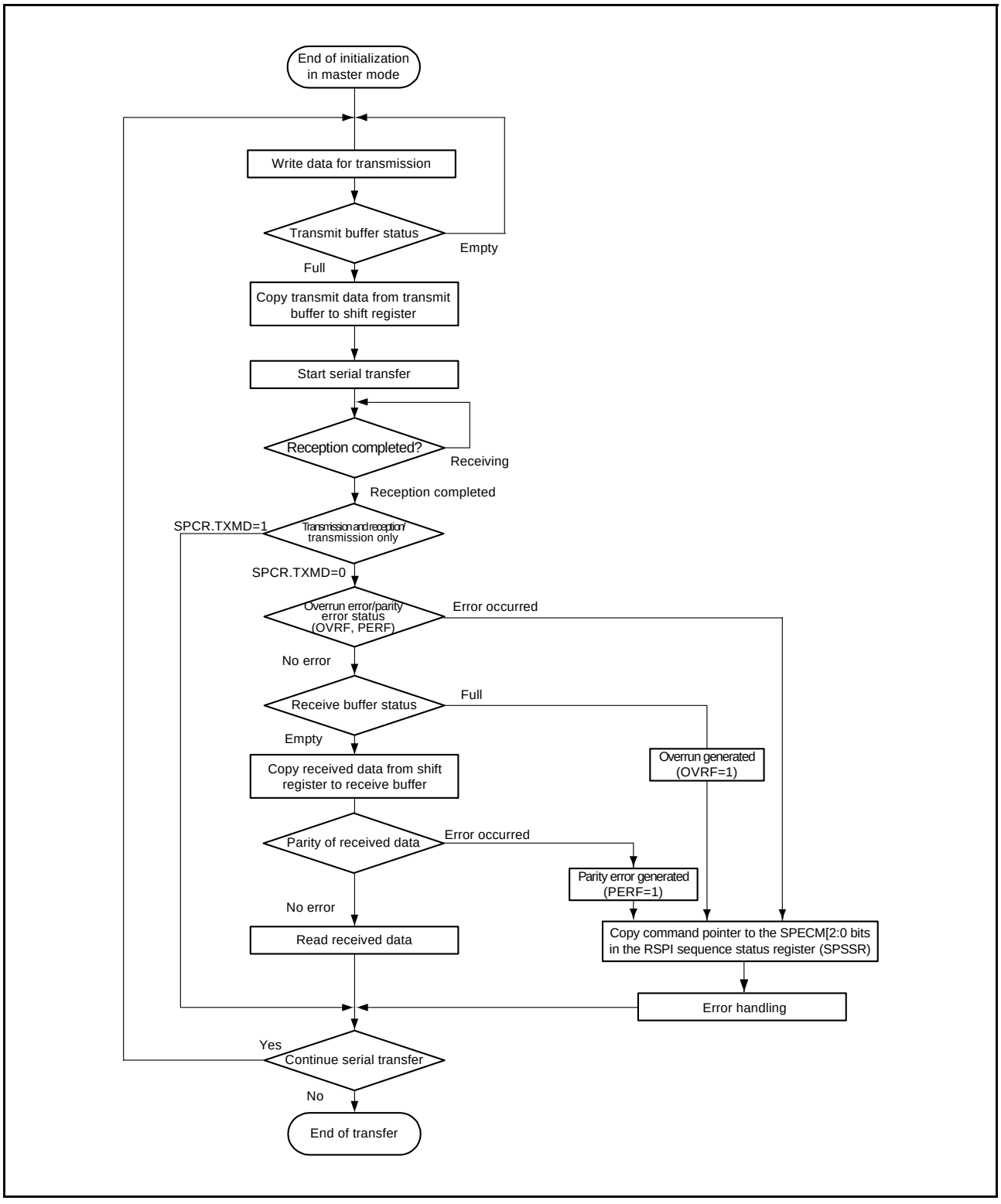


Figure 37.42 Example Flowchart for Transfer Operations in Master Mode (Clock Synchronous Operation)

37.3.13 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKn 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 always drives the MISOn output signal.
For details on the RSPI transfer format, see section 37.3.5, Transfer Format. It should be noted that the SSL0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing.
When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR). Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”. 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, see section 37.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 37.43 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, see the descriptions given in the individual blocks.

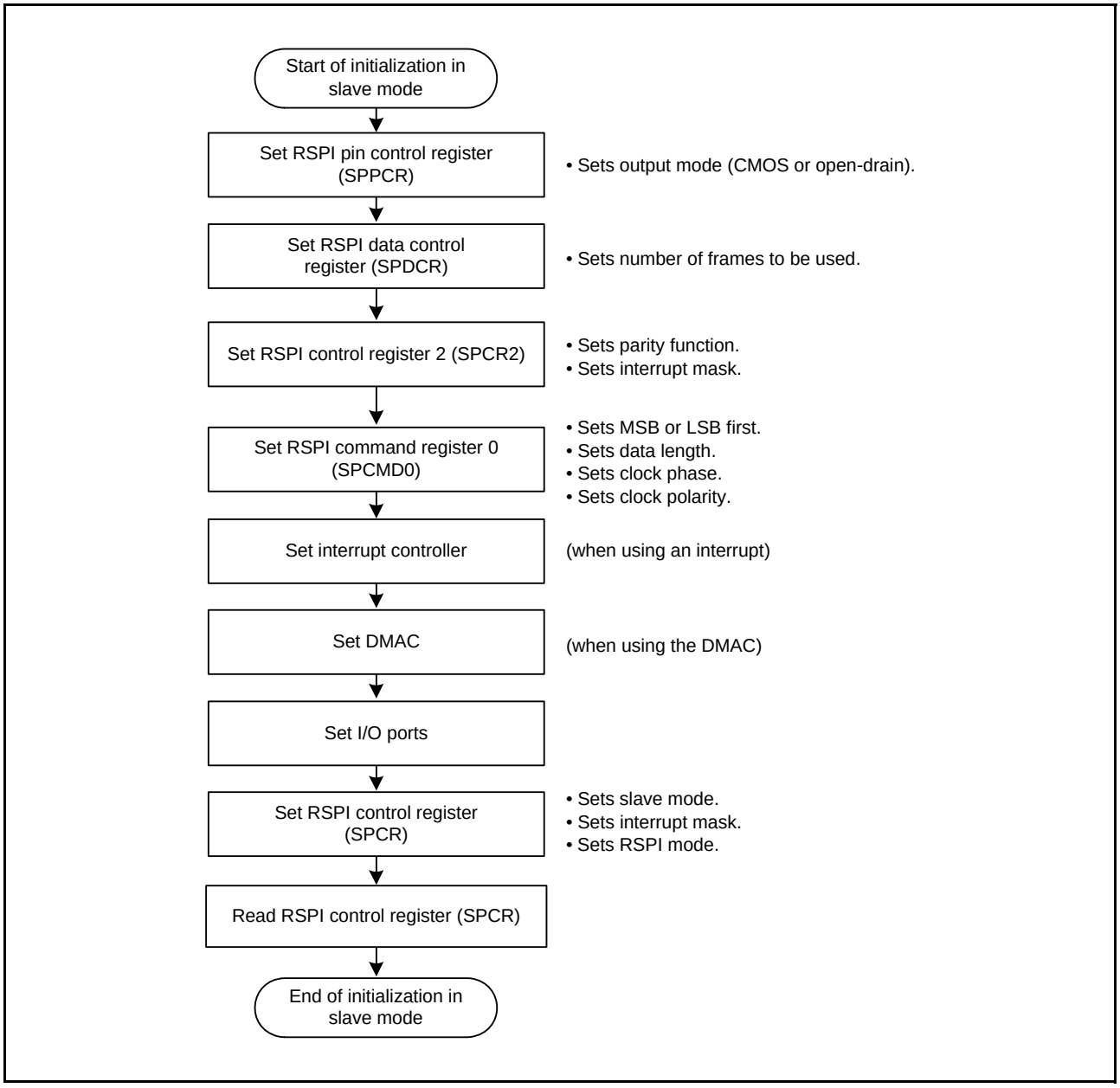


Figure 37.43 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Transfer Operation Flowcharts

Figure 37.44 is a flowchart illustrating clock synchronous operations in slave mode.

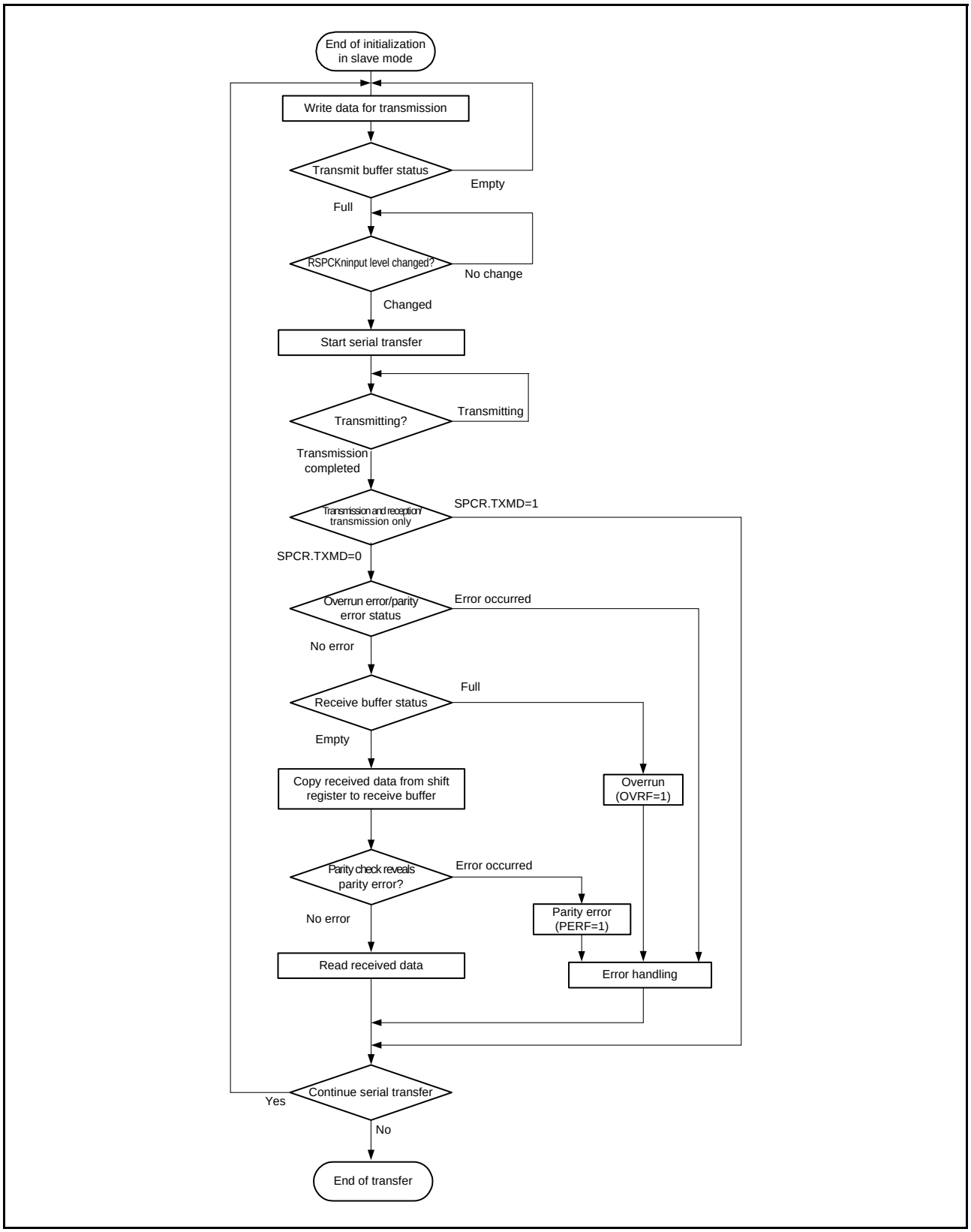


Figure 37.44 Example Flowchart for Transfer Operations in Slave Mode (CPHA = 1) (Clock Synchronous Operation)

37.3.14 Error Handling

Figure 37.45 to Figure 37.47 show error handling for the RSPI. The following error handling is used to return from the error state after an error has occurred in master mode or slave mode.

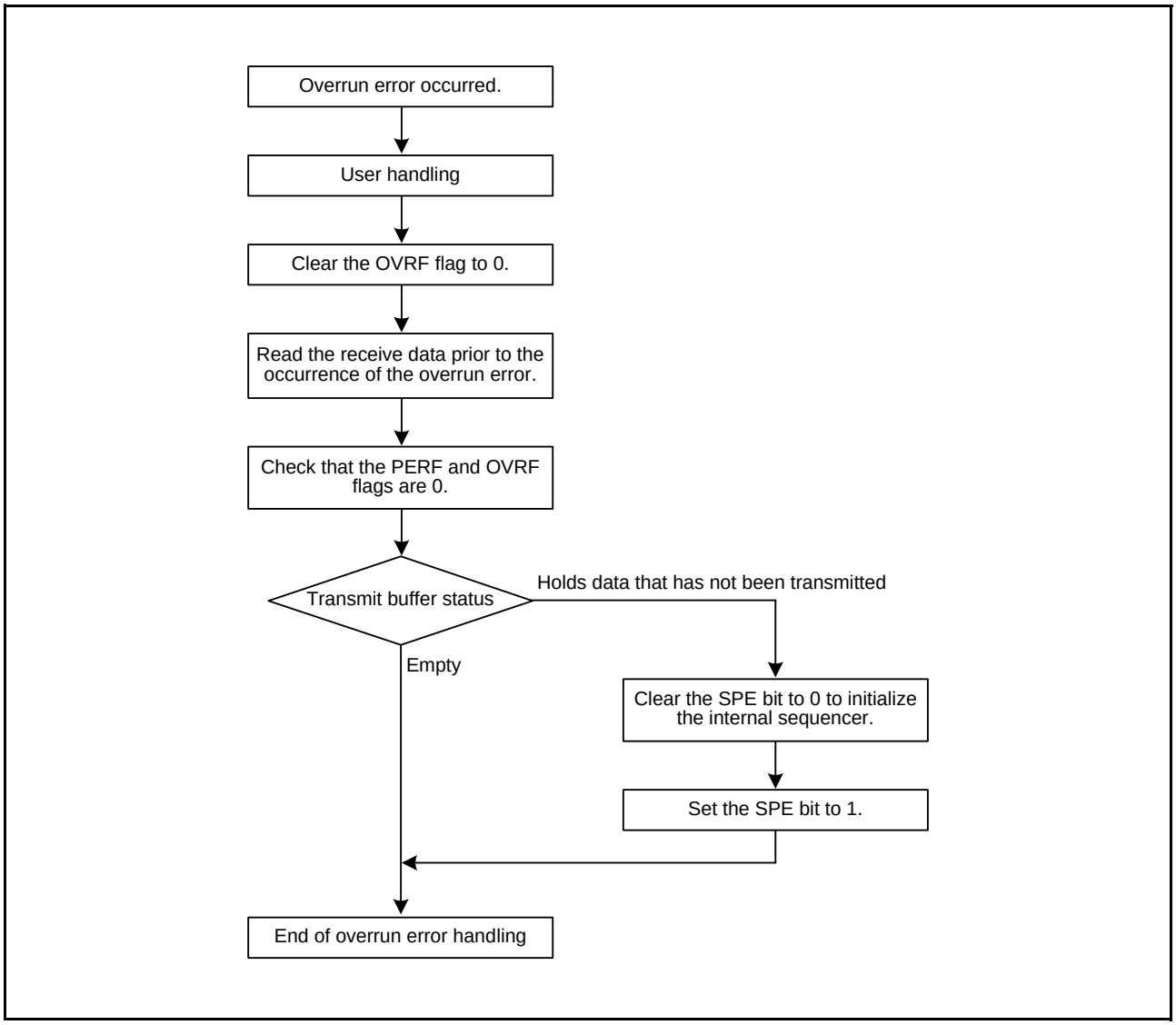


Figure 37.45 Error Handling (Overrun Error)

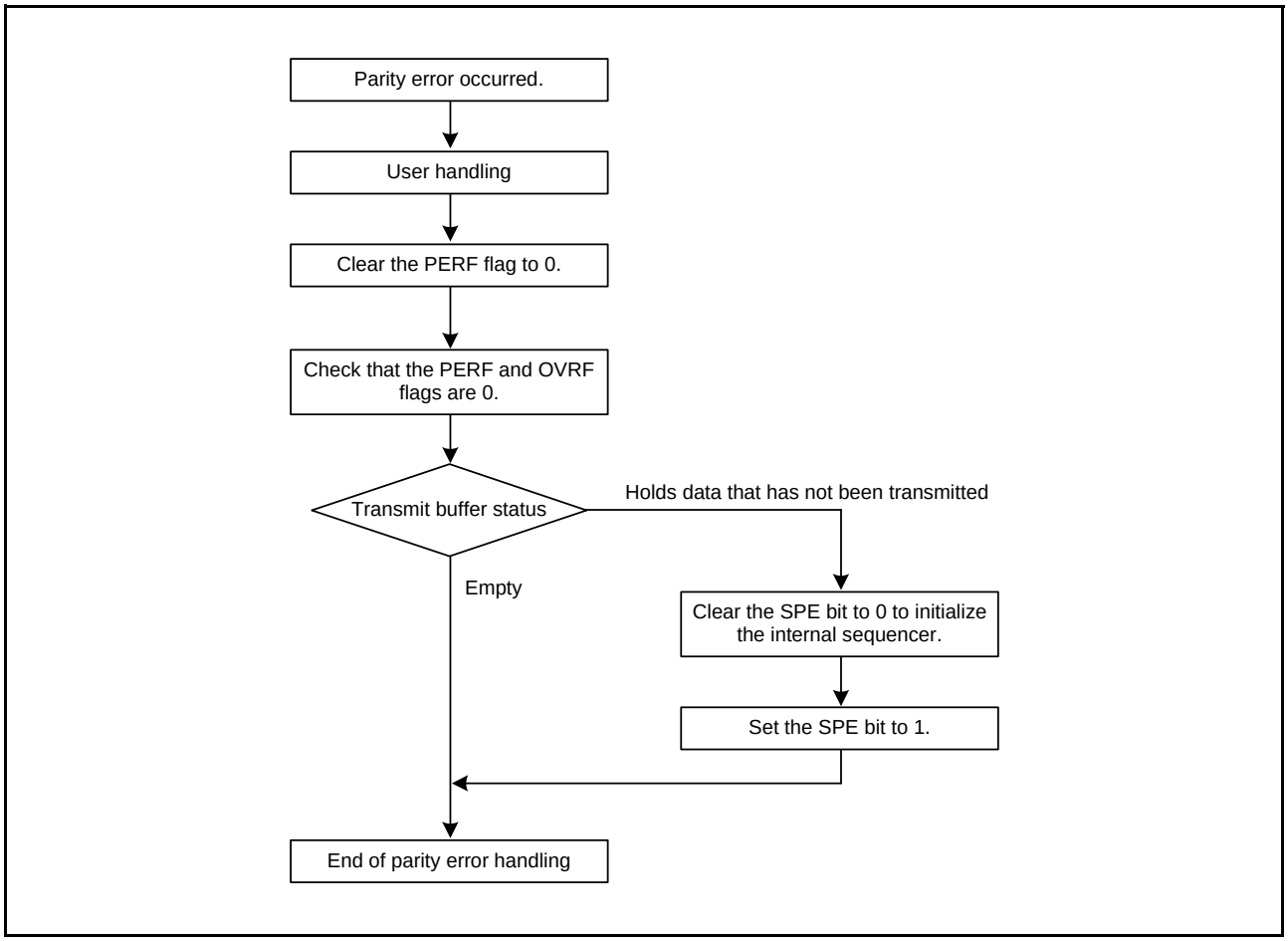


Figure 37.46 Error Handling (Parity Error)

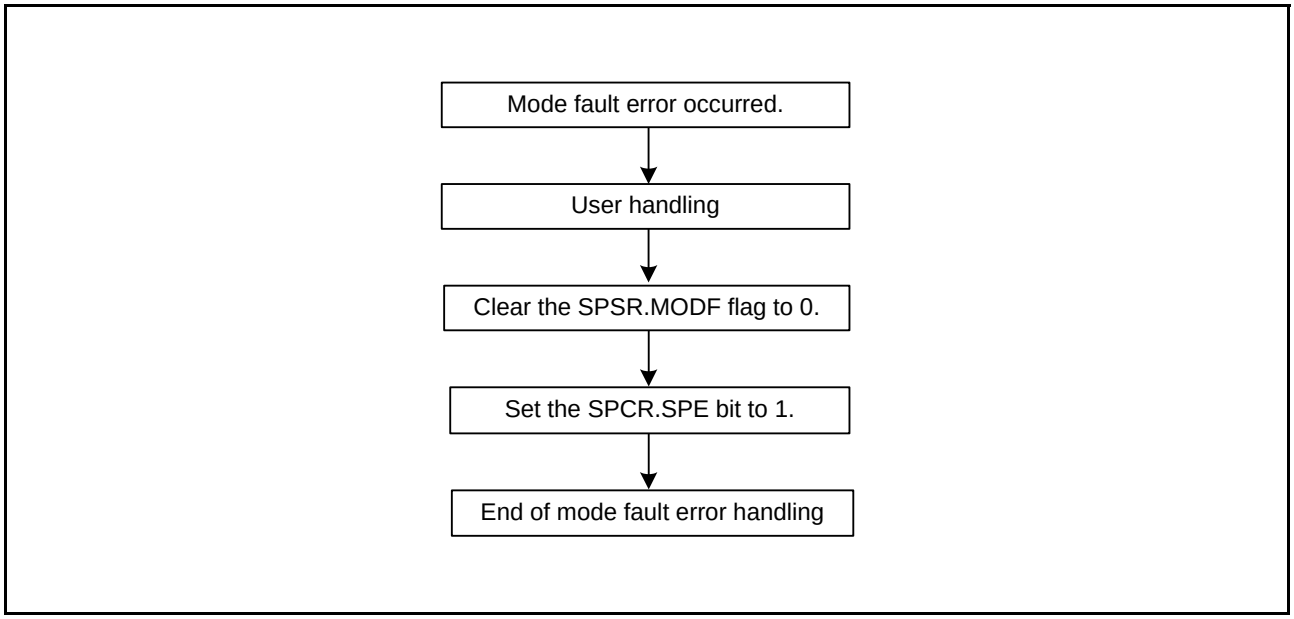


Figure 37.47 Error Handling (Mode Fault Error)

37.3.15 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn 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 MOSIn pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOn 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 37.12 lists the relationship among the SPLP2 and SPLP bits in SPPCR and the received data. Figure 37.48 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 37.12 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Reversed transmit data
1	0	Transmit data
1	1	Transmit data

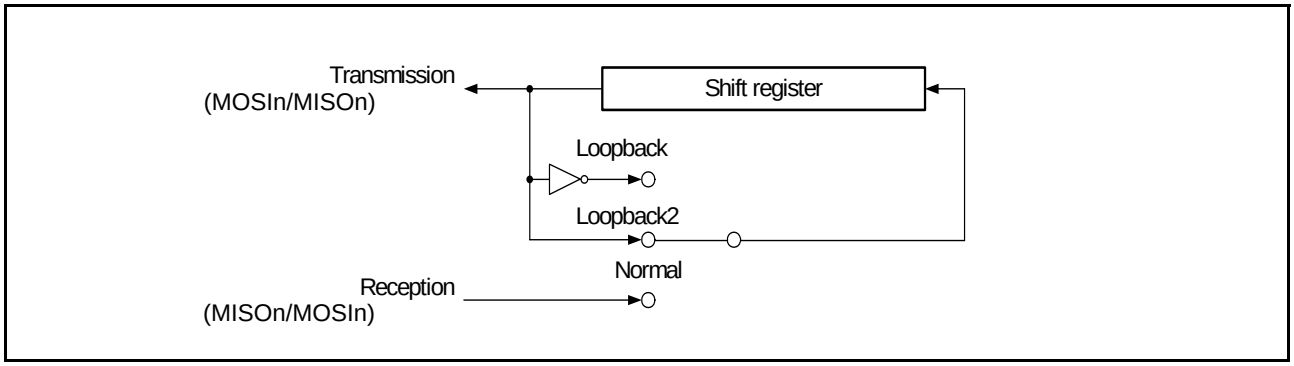


Figure 37.48 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

37.3.16 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 37.49.

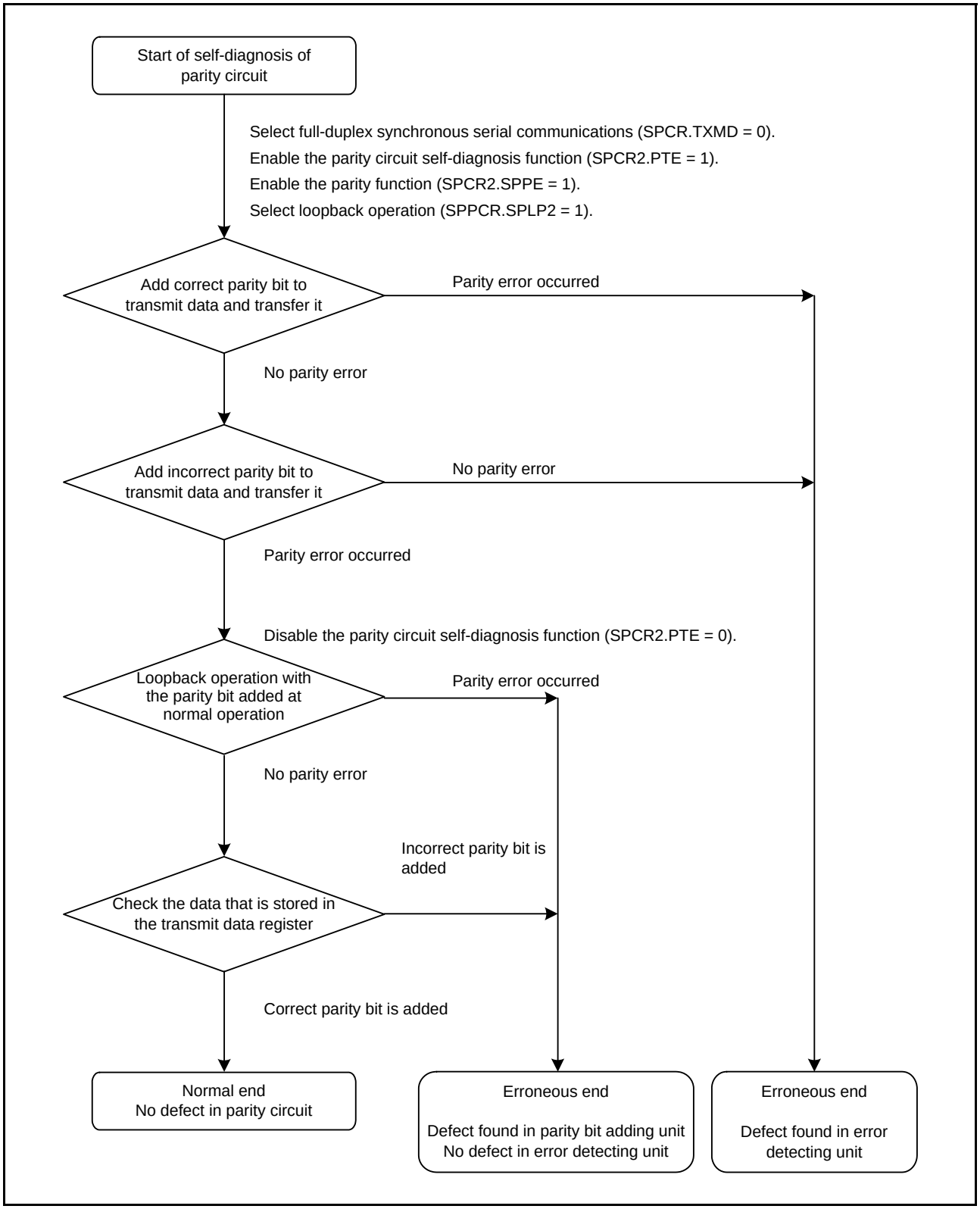


Figure 37.49 Flowchart for Self-Diagnosis of Parity Circuit

37.3.17 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 37.13. An interrupt is generated on satisfaction of an interrupt condition in Table 37.13. 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 17, DMA Controller (DMACA), or section 19, 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 to ICU 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 37.13 Interrupt Sources of RSPI

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full	SPRI	The receive buffer becomes full while the SPCR.SPRIE bit is 1.	Possible
Transmit buffer empty	SPTI	The transmit buffer becomes empty 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

37.4 Usage Note

37.4.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) or module stop control register C (MSTPCRC) can used to enable or disable operation of the RSPI. The RSPI 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.

37.4.2 Cautionary Note on the Low Power Consumption Functions

When a low power consumption function is to be used to lower power consumption by the RSPI, use the low power consumption function after the SPCR.SPE bit is set to 0 and transfer ends.

37.4.3 Points to Note 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 ICCR1.ICE bit 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.

38. IEBus Controller (IEB)

38.1 Overview

The RX63N/RX631 Group has an on-chip one-channel IEBus controller (IEB).
The Inter Equipment Bus™ (IEBus™)*1 implements a small-scale digital data transfer system for inter-equipment data transfer.
The RX63N/RX631 Group does not have an on-chip IEBus driver/receiver, so it is necessary to mount a dedicated driver/receiver externally.
Table 38.1 lists the specifications of the IEB, Figure 38.5 shows a block diagram of the IEB, and Table 38.8 lists the input/output pins of the IEB.
Note 1. IEBus™ (Inter Equipment Bus™) is a trademark of Renesas Electronics Corporation.

Table 38.1 Specifications of IEB

Item	Specifications
Communications protocol	• IEBus protocol control (layer 2) supported Half-duplex asynchronous communications Multi-master system Broadcast communications function Selectable two communications modes (modes 0, 1) with different transfer speeds
Data transmission and reception	• On-chip buffers for data transmission and reception Transmission and reception buffers: 32 bytes each Up to 32 bytes of consecutive transmit/reception (maximum number of transfer bytes in mode 1)
Operating frequency	12 MHz, 12.58 MHz (IEB uses 1/2 divided IECLK) 18 MHz, 18.87 MHz (IEB uses 1/3 divided IECLK) 24 MHz, 25.16 MHz (IEB uses 1/4 divided IECLK) 30 MHz, 31.45 MHz (IEB uses 1/5 divided IECLK) 36 MHz, 37.74 MHz (IEB uses 1/6 divided IECLK) 42 MHz, 44.03 MHz (IEB uses 1/7 divided IECLK)
Power consumption reducing function	Module stop state can be set.

38.1.1 IEBus Communications Protocol

- Communications method: Half-duplex asynchronous communications
- Multi-master system
  All units connected to the IEBus can transfer data to other units.
- Broadcast communications function (one-to-many communications)
  Group broadcast communications: Broadcast communications to group units
  General broadcast communications: Broadcast communications to all units
- Mode is selectable (two modes with different transfer speeds)

Table 38.2 Communications Modes

Mode	IECLK = 12 MHz, 18 MHz, 24 MHz, 30 MHz, 36 MHz, or 42 MHz	IECLK = 12.58 MHz, 18.87 MHz, 25.16 MHz, 31.45 MHz, 37.74 MHz, or 44.03 MHz	Maximum Number Of Transfer Bytes (byte/frame)
0	About 3.9 kbps	About 4.1 kbps	16
1	About 17 kbps	About 18 kbps	32

- Access control: CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
  Priority of bus mastership is as follows.
  Broadcast communications (one-to-many communications) have priority over normal communications (one-to-one communications).
  A smaller master address has priority.
- Communications scale*1
  Number of units: Up to 50
  Cable length: Up to 150 m (when using a twisted-pair cable)

Note 1. The communications scale of the actual system depends on the characteristics of the externally mounted IEBus driver/receiver and the cable used.

(1) Determination of Bus Mastership (Arbitration)

A unit connected to the IEBus performs an operation to get the bus to control other units. This operation is called arbitration. In arbitration, when multiple units start transferring simultaneously, the bus mastership is given to one unit among them.
Only one unit can obtain bus mastership through arbitration, so the following priority for bus mastership is determined.

(a) Priority according to communications type

Broadcast communications (one-to-many communications) has priority over normal communications (one-to-one communications).

(b) Priority according to master address

The unit with the smallest master address has priority among units of the same communications type.
Example: The master address is configured with 12 bits. A unit with 000h has the highest priority, while a unit with FFFh has the lowest priority.

Note: • When a unit loses in arbitration, the unit can automatically enter retransfer mode (0 to 7 retransfer times can be selected by setting the RN bit in IEMCR).

(2) Communications Mode

The IEBus has two communications modes with different transfer speeds. Table 38.3 lists the transfer speed in each communications mode and the maximum number of transfer bytes in one communications frame.

Table 38.3 Transfer Speed and Maximum Number of Transfer Bytes in Each Communications Mode

Communications Mode	Maximum Number of Transfer Bytes (bytes/frame)	Effective Transfer Speed*1 (kbps)	
		IECLK = 12 MHz, 18 MHz, 24 MHz, 30 MHz, 36 MHz, or 42 MHz	IEBCLK = 12.58 MHz, 18.87 MHz, 25.16 MHz, 31.45 MHz, 37.74 MHz, or 44.03 MHz
0	16	About 3.9	About 4.1
1	32	About 17	About 18

Note: • Each unit connected to the IEBus should select a communications mode prior to performing communications. Note that correct communications is not guaranteed if the master and slave units do not adopt the same communications mode.
In the case of communications between a unit with IECLK = 12 MHz and a unit with IECLK = 12.58 MHz, correct communications are not possible even if the same communications mode is adopted. Communications must be done with the same oscillation frequency.

Note 1. Effective transfer speed when the maximum number of transfer bytes is transmitted.

(3) Communications Address

In the IEBus, a specific 12-bit communications address is allocated to each individual unit. A communications address is configured as follows.
Upper four bits: Group number (number identifying a group to which the unit belongs)
Lower eight bits: Unit number (number identifying individual units in a group)

(4) Broadcast Communications

In normal transfer, a single master unit communicates with a single slave unit, so one-to-one transmission or reception takes place. In broadcast communications, a single master unit communicates with multiple slave units. Since there are multiple slave units, no acknowledgements are returned from the slave units during communications.
A broadcast bit decides whether broadcast or normal communications is done. (For details of the broadcast bit, see section 38.1.2 (1), (b) Broadcast Bit.)
There are two types of broadcast communications.

(a) Group broadcast communications

Broadcast communications is aimed at units with the same group number, meaning that those units have the same upper four bits of the communications address.

(b) General broadcast communications

Broadcast communications is aimed at all units regardless of group number.
Group broadcast and general broadcast communications are identified by a slave address. (For details on the slave address, see section 38.1.2, (3) Slave Address Field.)

38.1.2 Communications Protocol

Figure 38.1 shows an IEBus transfer signal format.
Communications data is transferred as a series of signals referred to as a communications frame. The number of data which can be transferred in a single communications frame and the transfer speed differ according to the communications mode.

When IECLK = 12 MHz, 18 MHz, 24 MHz, 30 MHz, 36 MHz, or 42 MHz

Field name	Header		Master address field		Slave address field			Control field			Message length field			Data field							
Number of bits	1	1	12	1	12	1	1	4	1	1	8	1	1	8	1	1		8	1	1
	Start bit	Broad-cast bit	Master address	P	Slave address	P	A	Control bits	P	A	Message length bits	P	A	Data bits	P	A		Data bits	P	A
Transfer time																					
Mode 0	Approximately 7330 µs													Approximately 1590 × N µs							
Mode 1	Approximately 2090 µs													Approximately 410 × N µs							

P: Parity bit (1 bit)
A: Acknowledge bit (1 bit)
When A = 0: ACK
When A = 1: NAK
N: Number of bytes

Note: • The value of acknowledge bit is ignored in broadcast communications.

Figure 38.1 Transfer Signal Format

(1) Header

A header is comprised of a start bit and a broadcast bit.

(a) Start Bit

The start bit is a signal to inform other units of the start of data transfer.
A unit attempting to start data transfer outputs a low-level signal (the start bit) for a specified period and then outputs the broadcast bit.
If another unit is already outputting a start bit when a unit attempts to output a start bit, the unit waits for completion of the start bit from the other unit without outputting its own start bit, and then outputs the broadcast bit synchronized with the completion timing.
Other units enter the receive state after detecting the start bit.

(b) Broadcast Bit

The broadcast bit is a bit to identify the type of communications: broadcast or normal.
When this bit is cleared to 0, it indicates broadcast communications. When it is set to 1, it indicates normal communications. Broadcast communications includes group broadcast and general broadcast, which are identified by a value of the slave address. (For details of the slave address, see section 38.1.2, (3) Slave Address Field.)
Since multiple slave units are communications destination units in the case of broadcast communications, the acknowledge bit is not returned from each field described in (2) and below.
When more than one unit starts to transfer a communications frame with the same timing, broadcast communications has priority over normal communications and gets the bus mastership through arbitration.

(2) Master Address Field

The master address field is a field for transmitting the unit address (master address) to other units. The master address field is comprised of master address bits and a parity bit.
The master address consists of 12 bits and the MSB is output first.
When more than one unit start to transfer broadcast bits having the same value with the same timing, arbitration is decided by the master address field.
In the master address field, self-output data and data on the bus are compared for every one-bit transfer. If the self-output master address and data on the bus are different, the unit loses arbitration, stops its transfer, and enters the receive state.
Since the IEBus is configured with wired AND, the unit having the smallest master address among the units in arbitration (arbitration master) wins in arbitration.
Finally, only a single unit remains in the transfer state as a master unit after outputting a 12-bit master address.
Next, this master unit outputs a parity bit*1, defines the master address for other units, and then enters the slave address field output state.

Note 1. Since even parity is used, when the number of bits having the value 1 in the master address is odd, the parity bit is 1.

(3) Slave Address Field

The slave address field is a field to transmit an address (the slave address) of a unit (the slave unit) to communicate with.
The slave address field is comprised of slave address bits, a parity bit, and an acknowledge bit.
The slave address consists of 12 bits and the MSB is output first. The parity bit is output after the 12-bit slave address is transmitted to avoid receiving an incorrect slave address. The master unit then detects the acknowledgement from the slave unit to confirm that the slave unit exists on the bus. When the acknowledgement is detected, the master unit enters the control field output state. However, the master unit enters the control field output state without detecting the acknowledgement in broadcast communications.
The slave unit returns an acknowledgement when the slave address matches its own address and the parities of the master and slave addresses are even. When the parity is odd, the slave unit decides that the master or slave address was not correctly received and does not return the acknowledgement. In this case, the master unit enters the waiting (monitor) state and communications ends.

In the case of broadcast communications, the slave address is used to identify the type of broadcast communications (group*1 or general) as follows:
When the slave address is FFFh: General broadcast communications
When the slave address is not FFFh: Group broadcast communications
Note 1. The group number is the upper 4-bit value of the slave address in group broadcast communications.

(4) Control Field

The control field is a field for transmitting the type and direction of the following data field. The control field is comprised of control bits, a parity bit, and an acknowledge bit.
The control bits consist of four bits and the MSB is output first.
The parity bit is output following the control bits. When the parity is even and the slave unit can execute the operation required from the master unit, the slave unit returns an acknowledgement and enters the message length field. However, if the slave unit cannot meet the requirements from the master unit even though the parity is even, or if the parity is odd, the slave unit does not return an acknowledgement and returns to the waiting (monitor) state.
The master unit enters the subsequent message length field after confirming the acknowledgement.
When the acknowledgement is not confirmed, the master unit enters the waiting (monitor) state, and communications ends. However, in the case of broadcast communications, the master unit enters the following message length field without confirming the acknowledgement.
For details of the contents of the control bit, see Table 38.5.

(5) Message Length Field

The message length field is a field for specifying the number of transfer bytes. The message length field is comprised of message length bits, a parity bit, and an acknowledge bit.
The message length bits consist of eight bits and the MSB is output first. Table 38.4 lists the number of transfer bytes.

Table 38.4 Contents of Message Length Bits

Message Length Bits	Number of Transfer Bytes
01h	1 byte
02h	2 bytes
:	:
:	:
20h	32 bytes
21h	Undefined
:	:
:	:
FFh	Undefined
00h	Undefined

Note: • If a number greater than the maximum number of transfer bytes in one frame is specified, communications are done in multiple frames depending on the communications mode. In this case, the message length bits indicate the number of remaining communications data after the first transfer. In the RX63N/RX631 Group, the value of the message length bits must be equal to or smaller than the maximum number of transfer bytes in one frame. Set the length within the ranges shown below.
Mode 0: 1 to 16 bytes
Mode 1: 1 to 32 bytes
If a number greater than the maximum number of transfer bytes is specified, the IEBus receives or transmits the first byte data without timing error and then terminates communications. After the end of communications, the receive-frame maximum transfer byte overflow bit in the IEBus receive status register or the transmit-frame maximum transfer byte overflow bit in the IEBus transmit status register is set.

This field operation differs depending on the value of bit 3 in the control field: master transmission (bit 3 of the control bits is 1) or master reception (bit 3 of the control bits is 0).

(a) Master Transmission

The master unit outputs the message length bits and the parity bit. When the parity is even, the slave unit returns an acknowledgement and enters the following data field. Note that the slave unit does not return an acknowledgement in broadcast communications.

When the parity is odd, the slave unit decides that the message length bits are not correctly received, does not return an acknowledgement, and returns to the waiting (monitor) state. In this case, the master unit also returns to the waiting state and communications end.

(b) Master Reception

The slave unit outputs the message length bits and the parity bit. When the parity is even, the master unit returns an acknowledgement.
When the parity is odd, the master unit decides that the message length bits are not correctly received, does not return an acknowledgement, and returns to the waiting state. In this case, the slave unit also returns to the waiting state and communications end.

(6) Data Field

The data field is a field for data transmission/reception to and from the slave unit. The master unit transmits/receives data to and from the slave unit using the data field. The data field is comprised of data bits, a parity bit, and an acknowledge bit.
The data bits consist of eight bits and the MSB is output first.
The parity and acknowledge bits are output following the data bits from the master unit and slave unit, respectively.
Broadcast communications are performed only for the transmission of the master unit. In this case, the acknowledge bit is ignored.
Operations in master transmission and master reception are described below.

(a) Master Transmission

The master unit transmits the data bits and parity bit to the slave unit to write data from the master unit to the slave unit. The slave unit receives the data bits and parity bit, and returns an acknowledgement if the parity bit is even and the receive buffer is empty. If the parity bit is odd or the receive buffer is not empty, the slave unit does not accept the corresponding data and does not return an acknowledgement.
When the slave unit does not return an acknowledgement, the master unit retransmits the data. This operation is repeated until either an acknowledgement from the slave unit is detected or the maximum number of data transfer bytes is reached.
When the parity is even and the acknowledgement is output from the slave unit, the master unit transmits the subsequent data if data remains and the maximum number of transfer bytes are not exceeded.
In the case of broadcast communications, the slave unit does not return the acknowledgement, and the master unit transfers data byte by byte.

(b) Master Reception

The master unit outputs synchronous signals corresponding to all data bits to be read from the slave unit.
The slave unit outputs the contents of the data bits and parity bit on the bus in accordance with the synchronous signals from the master unit.
The master unit reads the parity bit output from the slave unit, and checks the parity.
If the parity is odd, or the receive buffer is not empty, the master unit rejects acceptance of the data, and does not return the acknowledgement. The master unit reads the same data repeatedly if the number of data does not exceed the maximum number of transfer bytes in one frame. If the parity is even and the receive buffer is empty, the master unit accepts data and returns an acknowledgement. The master unit reads the subsequent data if the number of data does not exceed the maximum number of transfer bytes in one frame.

(7) Parity Bit

The parity bit is used to confirm that transfer data has no errors.
The parity bit is added to respective data of the master address, slave address, control, message length, and data bits.
Even parity is used. When the number of bits having the value 1 is odd, the parity bit is 1. When the number of bits having the value 1 is even, the parity bit is 0.

(8) Acknowledge Bit

In normal communications (single unit to single unit communications), the acknowledge bit is added in the following positions to confirm that data is correctly accepted.

- At the end of the slave address field
- At the end of the control field
- At the end of the message length field
- At the end of the data field

The acknowledge bit is defined below.

- 0: Indicates that the transfer data is acknowledged. (ACK)
- 1: Indicates that the transfer data is not acknowledged. (NAK)

Note that the acknowledge bit is ignored in the case of broadcast communications.

(a) Acknowledge bit at the End of the Slave Address Field

The acknowledge bit at the end of the slave address field becomes NAK in the following cases and transfer is stopped.

- When the parity of the master address or slave address bits is incorrect
- When a timing error (an error in bit format) occurs
- When there is no slave unit

(b) Acknowledge bit at the End of the Control Field

The acknowledge bit at the end of the control field becomes NAK in the following cases and transfer is stopped.

- When the parity of the control bits is incorrect
- When the bit 3 of the control bits is 1 (data write) although the slave receive buffer*1 is not empty
- When the control bits are set to data read (3h, 7h) although the slave transmit buffer*1 is empty
- When another unit which locked the slave unit requests 3h, 6h, 7h, Ah, Bh, Eh, or Fh in the control bits although the slave unit is locked
- When the control bits are the locked address read (4h, 5h) although the unit is not locked
- When a timing error occurs
- When the control bits are undefined

Note 1. See section 38.1.3, (1) Slave Status Read (Control Bits: 0h, 6h).

(c) Acknowledge Bit at the End of the Message Length Field

The acknowledge bit at the end of the message length field becomes NAK in the following cases and transfer is stopped.

- When the parity of the message length bits is incorrect
- When a timing error occurs

(d) Acknowledge Bit at the End of the Data Field

The acknowledge bit at the end of the data field becomes NAK in the following cases and transfer is stopped.

- When the parity of the data bits is incorrect*1
- When a timing error occurs after the previous transfer of the acknowledge bit
- When the receive buffer becomes full and cannot accept further data*1

Note 1. In this case, the data field is transferred repeatedly until the data reaches the maximum number of transfer bytes if the number of data does not exceed the maximum number of transfer bytes in one frame.

38.1.3 Transfer Data (Data Field Contents)

The data field contents are specified by the control bits.

Table 38.5 Control Bit Contents

Set Value	Bit 3*1	Bit 2	Bit 1	Bit 0	Function*2
0h	0	0	0	0	Reads slave status (SSR)
1h	0	0	0	1	Undefined
2h	0	0	1	0	Undefined
3h	0	0	1	1	Reads data and locks
4h	0	1	0	0	Reads locked address (lower 8 bits)
5h	0	1	0	1	Reads locked address (upper 4 bits)
6h	0	1	1	0	Reads slave status (SSR) and unlocks
7h	0	1	1	1	Reads data
8h	1	0	0	0	Undefined
9h	1	0	0	1	Undefined
Ah	1	0	1	0	Writes command and locks
Bh	1	0	1	1	Writes data and locks
Ch	1	1	0	0	Undefined
Dh	1	1	0	1	Undefined
Eh	1	1	1	0	Writes command
Fh	1	1	1	1	Writes data

Note 1. Depending on the value of bit 3 (MSB), the transfer directions of the message length bits in the following message length field and data in the data field vary.
When bit 3 is 1: Data is transferred from the master unit to the slave unit.
When bit 3 is 0: Data is transferred from the slave unit to the master unit.
Note 2. 3h, 6h, Ah, and Bh are control bits to specify lock setting and cancellation.
When the undefined values of 1h, 2h, 8h, 9h, Ch, and Dh are transmitted, the acknowledge signal is not returned.

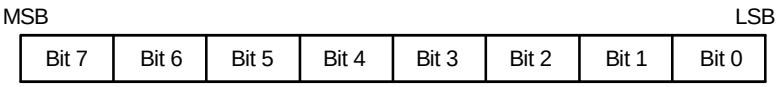
When the control bits received from a unit that is not the master unit that has requested a lock are not listed in Table 38.6, the slave unit locked by the master unit does not accept the control bits and does not return the acknowledge bit.

Table 38.6 Control Field for Locked Slave Unit

Set Value	Bit 3	Bit 2	Bit 1	Bit 0	Function
0h	0	0	0	0	Reads slave status
4h	0	1	0	0	Reads locked address (lower 8 bits)
5h	0	1	0	1	Reads locked address (upper 4 bits)

(1) Slave Status Read (Control Bits: 0h, 6h)

The master unit can check the reason the slave unit does not return the acknowledgement (ACK) by reading the slave status (0h, 6h). The slave status indicates the result of the last communications that the slave unit performed. All slave units can provide slave status information. Figure 38.2 shows the bit configuration of the slave status.



Bit	Value	Description	
Bits 7 and 6	00	Mode 0	Indicates the highest mode supported by the unit.*1
	01	Mode 1	
	10	For future use	
	11	For future use	
Bit 5	0	Fixed at 0	
Bit 4*2	0	Slave transmission stopped	
	1	Slave transmission enabled	
Bit 3	0	Fixed at 0	
Bit 2	0	Unit is unlocked	
	1	Unit is locked	
Bit 1*3	0	Slave receive buffer is empty	
	1	Slave receive buffer is not empty	
Bit 0*4	0	Slave transmit buffer is empty	
	1	Slave transmit buffer is empty	

Note 1. Since the RX63N/RX631 Group supports only modes 0 and 1 but not mode 2, bits 7 and 6 are fixed at 01.
Note 2. The value of bit 4 can be selected by the STE bit in the IEBus master unit address register 1 (IEAR1).
Note 3. The slave receive buffer is a buffer that is accessed during data write (control bits: Ah, Bh, Eh, Fh).
In the RX63N/RX631 Group, the slave receive buffer corresponds to the IEBus receive buffer registers (IERB001 to IERB032); and bit 1 is the value of the RXBSY bit in the IEBus receive status register (IERSR).
Note 4. The slave transmit buffer is a buffer that is accessed during data read (control bits: 3h, 7h).
In the RX63N/RX631 Group, the slave transmit buffer corresponds to the IEBus transmit buffer registers (IETB001 to IETB032) and bit 0 is the value of the SRQ bit in the IEBus general flag registers (IEFLG).

Figure 38.2 Bit Configuration of Slave Status (SSR)

(2) Data Command Transfer (Control Bits: Read (3h, 7h), Write (Ah, Bh, Eh, Fh))

In the case of data read (3h, 7h), data in the data buffer of the slave unit is read in the master unit. In the case of data write (Bh, Fh) or command write (Ah, Eh), data received in the slave unit is processed in accordance with the operation specification of the slave unit.

Note: • The user can select data and commands freely in accordance with the system.
Note: • 3h, Ah, or Bh may lock depending on the communications condition and status.

(3) Locked Address Read (Control Bits: 4h, 5h)

In the case of the locked address read (4h, 5h), the address (12 bits) of the master unit that issues the lock instruction is configured in bytes as shown below and read.

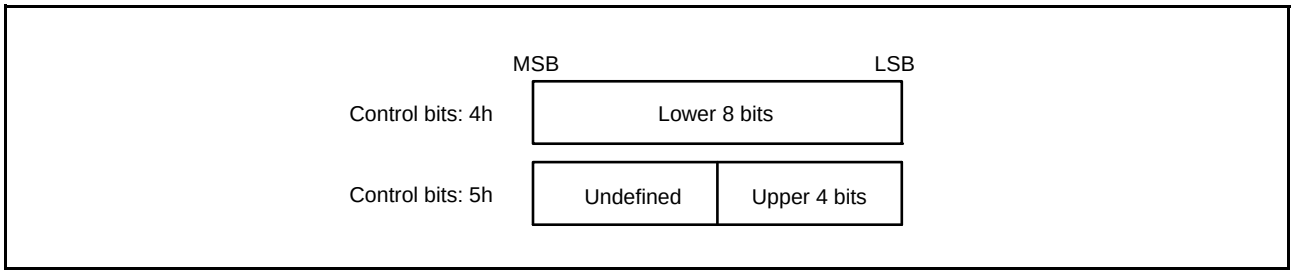


Figure 38.3 Locked Address Configuration

(4) Locking/Unlocking (Control Bits: Setting (3h, Ah, Bh), Cancellation: (6h))

The lock function is used for message transfer over multiple communications frames. A locked unit receives data only from the unit that locked it.
Locking and unlocking are described below.

(a) Locking

When an acknowledge bit of 0 in the message length field is transmitted/received with the control bits (3h, Ah, Bh) indicating the lock operation, and then the communications frame is completed before completion of data transmission/reception for the number of bytes specified by the message length bits, the slave unit is locked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte data indicating the slave status is set to 1.
Lock is set only when a transmit/receive-frame maximum transfer byte overflow is generated. Lock is not set by other error terminations.

(b) Unlocking

When the control bits indicate the lock (3h, Ah, Bh) or unlock (6h)) operation and the byte data for the number of bytes specified by the message length bits are transmitted/received in a single communications frame, the slave unit is unlocked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte indicating the slave status is cleared to 0.
Note that locking and unlocking are not done in broadcast communications.

Note: • There are three ways to cause a locked unit to unlock itself.
• Perform a hardware reset
• Perform a software reset
• Issue an unlock command through the IEBus command register (IECMR)
Note that the LCK flag in IEFLG can be used to check whether the unit is locked or unlocked.

38.1.4 Bit Format

Figure 38.4 shows the bit format (conceptual diagram) configuring the IEBus communications frame.

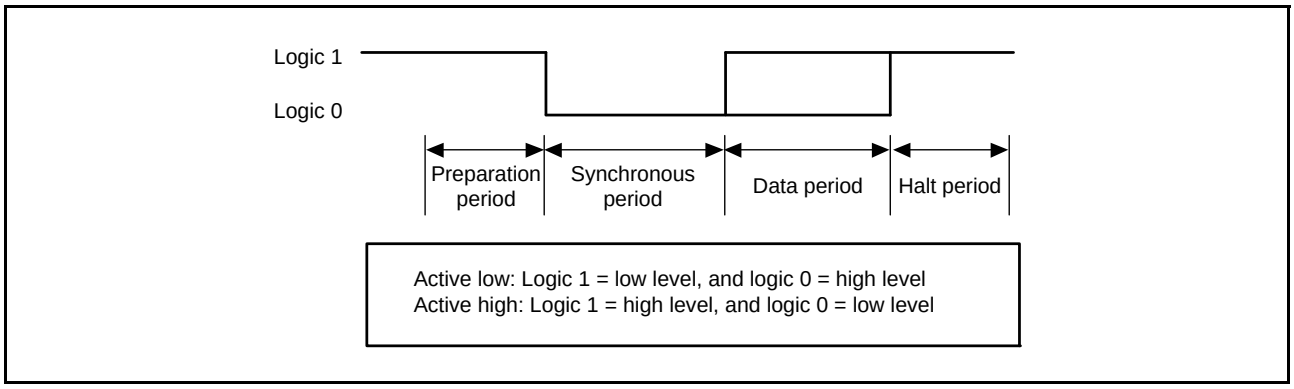


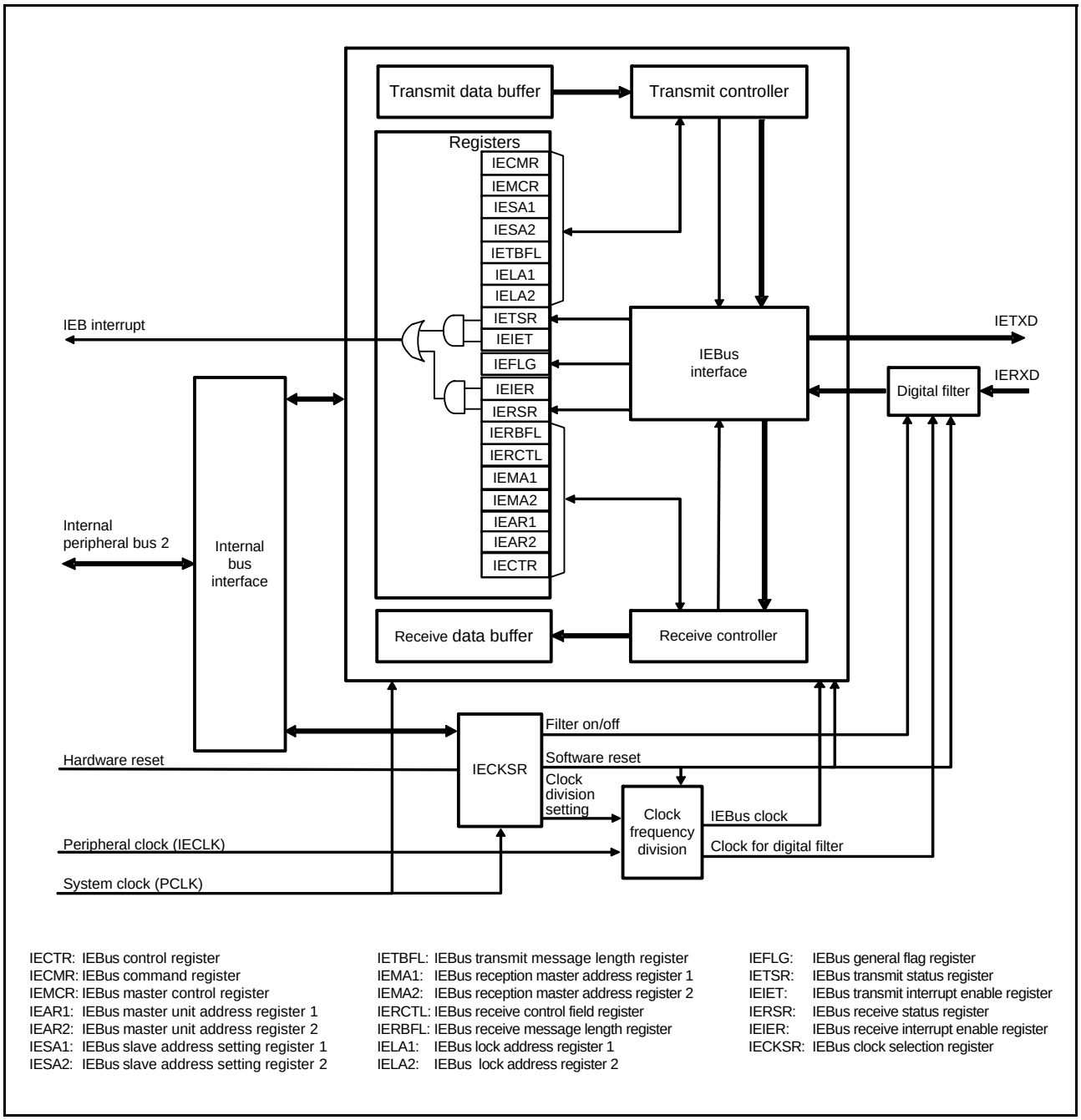
Figure 38.4 IEBus Bit Format (Conceptual Diagram)

Each period of the bit format for use of active high signals is described below.
Preparation period: first logic 1 period (high level)
Synchronous period: subsequent logic 0 period (low level)
Data period: period indicating the bit value (logic 1: high level, logic 0: low level)
Halt period: last logic 1 period (high level)

For use of active low signals, levels are reversed from the active high signals.
The synchronous and data periods have approximately the same length.
The IEBus is synchronized bit by bit. The specifications for the time of all bits and the periods allocated to the bits differ depending on the type of transfer bits and the unit (master or slave unit).

38.1.5 Block Diagram

Figure 38.5 shows the block diagram of the IEB and Table 38.7 lists the functions of each block.



IECTR: IEBus control register
IECMR: IEBus command register
IEMCR: IEBus master control register
IEAR1: IEBus master unit address register 1
IEAR2: IEBus master unit address register 2
IESA1: IEBus slave address setting register 1
IESA2: IEBus slave address setting register 2

IETBFL: IEBus transmit message length register
IEMA1: IEBus reception master address register 1
IEMA2: IEBus reception master address register 2
IERCTL: IEBus receive control field register
IERBFL: IEBus receive message length register
IELA1: IEBus lock address register 1
IELA2: IEBus lock address register 2

IEFLG: IEBus general flag register
IETSR: IEBus transmit status register
IEIET: IEBus transmit interrupt enable register
IERSR: IEBus receive status register
IEIER: IEBus receive interrupt enable register
IECKSR: IEBus clock selection register

Figure 38.5 Block Diagram of IEB

Table 38.7 Functions of Each Block

Block	Function
Bus interface	Internal bus interface
	• Data width: 8 bits • IEB register access
IEBus interface	Interface conforms to IEBus specifications
	• Outputs data from transmit controller to IEBus in IEBus specification bit format • Picks out frame data in IEBus specification bit format to transfer to receive controller
IECKSR	Division of the clock for IEBus/digital filter, selected by the IEBus clock selection register Software reset setting function
	• Assertion/negation of the asynchronous software reset for all FFs except the IECKSR register • Digital filter on/off setting • Clock frequency division setting of the input clock to the IEBus clock • Clock frequency division setting of the input clock to the digital filter clock
Registers	IEB control registers
	• Registers to control IEB • Readable/writable from internal bus
Clock division	Division output function of the input clock
	• Divides the input clock to be used as the IEBus clock, and outputs it to the IEBus interface • Divides the input clock to be used as the digital filter clock, and outputs it to the digital filter
Digital filter	Filtering function of the input to the IERXD pin of the IEBus
	• Digital filter on/off setting • Three-strand match filtering
Transmit controller	Transmits data in transmit buffer to IEBus
	• Generates transmit frame combining header information in register and data in transmit buffer to transmits • Detects transmit error
Receive controller	Stores data from IEBus in receive buffer
	• Stores header information and data in received frame in register and receive buffer, respectively • Detects receive error
Transmit data buffer	Buffer for data transmission
	• Buffer that stores data to be transmitted to IEBus • Buffer size: 32 bytes
Receive data buffer	Buffer for data reception
	• Buffer that stores data received from IEBus • Buffer size: 32 bytes

Table 38.8 Input/Output Pins of IEB

Pin Name	I/O	Function
IERXD	Input	Receive data input pin
IETXD	Output	Transmit data output pin

38.2 Register Descriptions

38.2.1 IEBus Control Register (IECTR)

Address(es): 0008 A800h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	IOL	DEE	—	RE	—	—	—
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
b3	RE	Receive Enable	0: Reception is disabled. 1: Reception is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DEE	Broadcast Receive Error Interrupt Enable	0: A broadcast receive error is not generated up to the control field. 1: A broadcast receive error is generated up to the control field.	R/W
b6	IOL	Input/Output Level	0: Pin input/output is set to active low. (Logic 1 is low level and logic 0 is high level.) 1: Pin input/output is set to active high. (Logic 1 is high level and logic 0 is low level.)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

RE Bit (Receive Enable)

This bit enables or disables IEB reception. The RE bit must be set at the initial setting before frame reception.

DEE Bit (Broadcast Receive Error Interrupt Enable)

If the DEE bit is set to 1, a broadcast reception error interrupt occurs when the receive buffer is not in the receive enabled state during broadcast reception (when the RE bit is not set to 1 or the IERSR.RXBSY flag is set). At this time, the master address is stored in IEBus reception master address registers 1 and 2.

While the DEE bit is 0, a broadcast reception error interrupt does not occur when the receive buffer is not in the receive enabled state, and the reception stops and enters the wait state. The master address is not saved.

IOL Bit (Input/Output Level)

This bit selects the input/output pin level (polarity) for the IERXD and IETXD pins.

38.2.2 IEBus Command Register (IECMR)

Address(es): 0008 A801h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CMD[2:0]		
Value after reset:	x	x	x	x	x	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD[2:0]	Command	b2 b1 b0 0 0 0: No operation. Operation is not affected. 0 0 1: Cancels the lock required from other units*1 0 1 0: Requires communications as the master 0 1 1: Stops master communications*2 1 0 0: Undefined*3 1 0 1: Requires data transfer from the slave 1 1 0: Stops data transfer from the slave*4 1 1 1: Undefined*3	W
b7 to b3	—	Reserved	The read value is undefined. The write value should be 0.	W

Note 1. Do not execute this command in slave communications.
Note 2. This command is valid only during master communications (IEFLG.MRQ = 1). In other states, this command issuance is ignored. If this command is issued in master communications, the communications controller immediately enters the wait state. At this time, the issued master transmission request ends (IEFLG.MRQ = 0).
Note 3. Undefined bits. Issuing this command does not affect operation.
Note 4. This command is valid only during slave communications (IEFLG.SRQ = 1). In other states, this command issuance is ignored. Once this command is issued in slave transmission, the IEFLG.SRQ flag is 0 before slave transmission. Therefore, a transmit request from the master is not responded to. If a transmit request is issued during slave transmission, the transmission stops and the wait state is entered (IEFLG.SRQ = 0).

Since this register is a write-only register, the read value is undefined.

CMD[2:0] Bits (Command)

These bits issue a command to control IEB communications. While the IEFLG.CMX flag is set after the command issuance, the command is in execution. When the IEFLG.CMX flag becomes 0, the operation state is entered.

38.2.3 IEBus Master Control Register (IEMCR)

Address(es): 0008 A802h

	b7	b6	b5	b4	b3	b2	b1	b0
	SS	RN[2:0]			CTL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CTL[3:0]*1	Control	b3 b2 b1 b0 0 0 0 0: Reads slave status 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: Reads data and locks*2 0 1 0 0: Reads locked address (lower 8 bits) 0 1 0 1: Reads locked address (upper 4 bits) 0 1 1 0: Reads slave status and unlocks*2 0 1 1 1: Reads data 1 0 0 0: Setting prohibited 1 0 0 1: Setting prohibited 1 0 1 0: Writes command and locks*2 1 0 1 1: Writes data and locks*2 1 1 0 0: Setting prohibited 1 1 0 1: Setting prohibited 1 1 1 0: Writes command 1 1 1 1: Writes data	R/W
b6 to b4	RN[2:0]	Retransmission Count Setting	b6 b5 b4 0 0 0: 0 0 0 1: 1 0 1 0: 2 0 1 1: 3 1 0 0: 4 1 0 1: 5 1 1 0: 6 1 1 1: 7	R/W
b7	SS	Broadcast/Normal Communications Select	0: Broadcast communications for master communications 1: Normal communications for master communications	R/W

Note 1. CTL[3] determines the data transfer direction of the message length bits in the message length field and data bits in the data field:
CTL[3] = 1: Transfer is from master unit to slave unit
CTL[3] = 0: Transfer is from slave unit to master unit
Note 2. Control bits for lock and unlock

IEMCR sets the communication conditions for master communications.

CTL[3:0] Bits (Control)

These bits set the control bits in the control field for master transmission.

RN[2:0] Bit (Retransmission Count Setting)

These bits set the number of times retransmission is done when arbitration is lost in master communications. If arbitration is lost, the TXEAL flag in IETSR is set and transmission ends.

38.2.4 IEBus Master Unit Address Register 1 (IEAR1)

Address(es): 0008 A803h

	b7	b6	b5	b4	b3	b2	b1	b0
	IARL4[3:0]				IMD[1:0]		—	STE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STE	Slave Transmission Setting	0: Bit 4 in the slave status register is 0 (slave transmission stop state) 1: Bit 4 in the slave status register is 1 (slave transmission enabled state)	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	IMD[1:0]	IEBus Communications Mode	b3 b2 0 0: Communications mode 0 0 1: Communications mode 1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b4	IARL4[3:0]	Lower 4 Bits of IEBus Master Unit Address	These bits set the lower four bits of the IEBus master unit address.	R/W

In master communications, the master unit address becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

STE Bit (Slave Transmission Setting)

This bit sets bit 4 in the slave status register. Transmitting the slave status register informs the master unit that the slave transmission enabled state is entered by setting the STE bit to 1. Note that STE bit only sets the slave status register value and does not directly affect slave transmission.

IARL4[3:0] Bits (Lower 4 Bits of IEBus Master Unit Address)

These bits set the lower four bits of the master unit address. This value becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

38.2.5 IEBus Master Unit Address Register 2 (IEAR2)

Address(es): 0008 A804h

	b7	b6	b5	b4	b3	b2	b1	b0
	IARU8[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IARU8[7:0]	Upper 8 Bits of IEBus Master Unit Address	These bits set the upper eight bits of the IEBus master unit address.	R/W

In master communications, this value becomes the master address field value. In slave communications, this value is compared with the received slave address field.

IARU8[7:0] Bits (Upper 8 Bits of IEBus Master Unit Address)

These bits set the upper eight bits of the master unit address. This value becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

38.2.6 IEBus Slave Address Setting Register 1 (IESA1)

Address(es): 0008 A805h

	b7	b6	b5	b4	b3	b2	b1	b0
	ISAL4[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 read as 0. The write value should be 0.	R/W
b7 to b4	ISAL4[3:0]	Lower 4 Bits of IEBus Slave Address	These bits set the lower four bits of IEBus slave address.	R/W

ISAL4[3:0] Bits (Lower 4 Bits of IEBus Slave Address)

These bits set the lower four bits of the communication destination slave unit address.

38.2.7 IEBus Slave Address Setting Register 2 (IESA2)

Address(es): 0008 A806h

	b7	b6	b5	b4	b3	b2	b1	b0
	ISAU8[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ISAU8[7:0]	Upper 8 Bits of IEBus Slave Address	These bits set the upper eight bits of the IEBus slave address.	R/W

ISAU8[7:0] Bits (Upper 8 Bits of IEBus Slave Address)

These bits set the upper eight bits of the communications destination slave unit address.

38.2.8 IEBus Transmit Message Length Register (IETBFL)

Address(es): 0008 A807h

	b7	b6	b5	b4	b3	b2	b1	b0
	IBFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IBFL[7:0]	Transmit Message Length	01h: 1 byte 02h: 2 bytes : : 1Fh: 31 bytes 20h: 32 bytes 21h: Setting prohibited : : FFh: Setting prohibited 00h: Setting prohibited	R/W

IETBFL sets the message length for master or slave transmission.

IBFL[7:0] Bits (Transmit Message Length)

These bits set the message length for master transmission.
Set the message length that does not exceed the maximum transmit bytes in communications mode.

38.2.9 IEBus Reception Master Address Register 1 (IEMA1)

Address(es): 0008 A809h

	b7	b6	b5	b4	b3	b2	b1	b0
	IMAL4[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 read as 0 and cannot be modified.	R
b7 to b4	IMAL4[3:0]	Lower 4 Bits of IEBus Reception Master Address	These bits indicate the lower four bits of the master unit address in IEBus slave/broadcast reception.	R

IMAL4[3:0] Bits (Lower 4 Bits of IEBus Reception Master Address)

These bits indicates the lower four bits of the communication destination master unit address in slave/broadcast reception. IEMA1 is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the IERSR.RXS flag.
If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the lower four bits of the master address are stored in IEMA1.

38.2.10 IEBus Reception Master Address Register 2 (IEMA2)

Address(es): 0008 A80Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	IMAU8[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IMAU8[7:0]	Upper 8 Bits of IEBus Reception Master Address	These bits indicate the upper eight bits of the master unit address in slave/broadcast reception.	R

IEMA2 is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the IERSR.RXS flag.
If a broadcast receive error interrupt is selected with the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper eight bits of the master address are stored in IEMA2.

IMAU8[7:0] Bits (Upper 8 Bits of IEBus Reception Master Address)

These bits indicate the upper eight bits of the communications destination master unit address in slave/broadcast reception. IEMA2 is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the IERSR.RXS flag.
If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper eight bits of the master address are stored in IEMA2.

38.2.11 IEBus Receive Control Field Register (IERCTL)

Address(es): 0008 A80Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RCTL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	RCTL[3:0]	IEBus Receive Control Field	These bits indicate the control field value in slave/broadcast reception.	R
b7 to b4	—	Reserved	These bits are read as 0 and cannot be modified.	R

IERCTL is enabled when slave/broadcast receive starts, and the contents are changed at the time of setting the RXS flag in IERSR.

RCTL[3:0] Bits (IEBus Receive Control Field)

These bits indicate the control field value in slave/broadcast reception. IERCTL is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the IERSR.RXS flag.

38.2.12 IEBus Receive Message Length Register (IERBFL)

Address(es): 0008 A80Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	RBFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RBFL[7:0]	IEBus Receive Message Length	These bits store the contents of the message length field in slave/ broadcast reception.	R

IERBFL is enabled when slave/broadcast receive starts, and the contents are changed at the time of setting the RXS flag in IERSR.

38.2.13 IEBus Lock Address Register 1 (IELA1)

Address(es): 0008 A80Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ILAL8[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ILAL8[7:0]	Lower 8 Bits of IEBus Locked Address	These bits indicate the lower eight bits of the master unit address when a unit is locked. These bits are valid only when the LCK flag in IEFLG is set.	R

38.2.14 IEBus Lock Address Register 2 (IELA2)

Address(es): 0008 A80Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—				ILAU4[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ILAU4[3:0]	Upper 4 Bits of IEBus Locked Address	These bits store the upper four bits of the master unit address when a unit is locked. These bits are valid only when the LCK flag in IEFLG is set.	R
b7 to b4	—	Reserved	These bits are read as 0 and cannot be modified.	R

38.2.15 IEBus General Flag Register (IEFLG)

Address(es): 0008 A810h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMX	MRQ	SRQ	SRE	LCK	—	RSS	GG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GG	General Broadcast Reception Acknowledgement Flag	0: A unit is in slave reception When FFFh is not acknowledged in the slave address field in broadcast reception 1: When FFFh is acknowledged in the slave address field in broadcast reception	R
b1	RSS	Receive Broadcast Flag	0: Received broadcast bit is 0 1: Received broadcast bit is 1	R
b2	—	Reserved	This bit is read as 0 and cannot be modified.	R
b3	LCK	Lock Status Indication Flag	0: A unit is unlocked 1: A unit is locked	R
b4	SRE	Slave Receive Status Flag	0: Slave/broadcast reception is not being executed 1: Slave/broadcast reception is being executed	R
b5	SRQ	Slave Transmission Request Flag	0: The unit is not in the transmit request state as a slave unit 1: The unit is in the transmit request state as a slave unit	R
b6	MRQ	Master Communications Request Flag	0: The unit is not in the communications request state as a master unit 1: The unit is in the communications request state as a master unit	R
b7	CMX	Command Execution Status Flag	0: Command execution is completed 1: A command is being executed	R

GG Flag (General Broadcast Reception Acknowledgement)

This flag is set to 1 when the slave address is acknowledged as FFFh in broadcast reception. Like the receive broadcast bit, this flag is valid when the slave/broadcast reception is started. (This flag is changed at the time of setting the RXS flag in IERSR.)

The previous value remains unchanged until the next slave/broadcast reception is started. This flag is 0 in slave normal reception.

RSS Flag (Receive Broadcast)

This flag indicates the received broadcast bit value. This flag is valid when the slave/broadcast reception is started. (This flag is changed at the time of setting the RXS flag in IERSR.)

The previous value remains unchanged until the next slave/broadcast reception is started.

LCK Flag (Lock Status Indication)

This flag is set to 1 when a unit is locked by a lock request from the master unit. The IELA1 and IELA2 values are valid only when the LCK flag is set to 1.

[Setting condition]

- When data for the number of bytes specified by the message length is not received/transmitted after the control bits (3h, Ah, Bh) that make the unit locked are received from the master unit. (The LCK flag is set to 1 only when a transmit/receive-frame maximum transfer byte overflow is generated. This flag is not set by completion with other errors.)

[Clearing condition]

- When an unlock condition is satisfied or when an unlock command is issued.

SRE Flag (Slave Receive Status)

This flag indicates the execution status in slave/broadcast reception.

[Setting condition]

- When the slave/broadcast reception is started while the RE bit in IECTR is set to 1.

[Clearing condition]

- When the slave/broadcast reception has been completed.

SRQ Flag (Slave Transmission Request)

This flag indicates whether the unit is in the transmit request state as a slave unit.

[Setting condition]

- When the CMX flag is cleared to 0 after the slave transmit request command is issued.

[Clearing condition]

- When a slave transmission has been completed.

MRQ Flag (Master Communications Request)

This flag indicates whether the unit is in the communications request state as a master unit.

[Setting condition]

- When the CMX flag is cleared to 0 after the master communications request command is issued.

[Clearing condition]

- When the master communications have been completed.

CMX Flag (Command Execution Status)

This flag indicates the command execution status.

[Setting condition]

- When a master communications request or slave transmit request command is issued while the MRQ, SRQ, or SRE flag is set.

[Clearing condition]

- When a command execution has been completed.

38.2.16 IEBus Transmit Status Register (IETSR)

Address(es): 0008 A811h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TXS	TXF	—	TXEAL	TXETTME	TXERO	TXEACK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXEACK	Acknowledge Flag	0: 1 (NAK) has not been detected in the acknowledge bit. 1: 1 (NAK) has been detected in the acknowledge bit.	R/(W) *1
b1	TXERO	Transmit-Frame Maximum Transfer Byte Overflow Flag	0: Transmit has been completed after the maximum number of bytes defined by the communications mode have been transmitted. 1: Transmit has not been completed although the maximum number of bytes defined by the communications mode have been transmitted.	R/(W) *1
b2	TXETTME	Transmit Timing Error Flag	0: No timing error has occurred during data transmission. 1: A timing error has occurred during data transmission.	R/(W) *1
b3	TXEAL	Arbitration Loss Flag	0: Normal transmission 1: Arbitration has been lost during data transmission and the transmission has been terminated.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TXF	Transmit Normal Completion Flag	0: Data has not been transmitted for the number of bytes specified by the message length bits. 1: Data has been transmitted for the number of bytes specified by the message length bits.	R/(W) *1
b6	TXS	Transmit Start Flag	0: The master address field transmission has not been completed in master transmission. 1: Arbitration has been won and the master address field transmission has been completed in master transmission.	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to clear the flag.

Each status flag in IETSR corresponds to a bit in the IEBus transmit interrupt enable register (IEIET) that enables or disables each interrupt. IETSR is cleared by writing 1 to each bit.

TXEACK Flag (Acknowledge)

This flag indicates the data received in the acknowledge bit of the data field.

- Acknowledge bit other than in the data field
  The IEB terminates the transmission and enters the wait state if a NAK is received. In this case, the TXEACK flag is set to 1.
- Acknowledge bit in the data field
  The IEB retransmits data up to the maximum number of bytes defined by the communications mode until an ACK is received from the receive unit if a NAK is received from the receive unit during data field transmission. In this case, when an ACK is received from the receive unit during retransmission, the TXEACK flag is not set and transmission will be continued. When transmission is terminated without receiving an ACK within the maximum number of bytes defined by the communications mode, the TXEACK flag is set to 1.

Note: • The TXEACK flag is invalid in broadcast communications.

[Setting condition]

- When the acknowledge bit of 1 (NAK) is detected.

[Clearing condition]

- When 1 is written.

TXERO Flag (Transmit-Frame Maximum Transfer Byte Overflow)

This flag indicates that the maximum number of bytes defined by the communications mode have been transmitted because a NAK has been received from the receive unit and retransmit has been performed, or that transmission has not been completed because the message length value exceeds*1 the maximum number of transfer bytes in one frame. The IEB sets the TXERO flag and enters the wait state.

Note 1. When the value specified as the message length is greater than the maximum number of transfer bytes, the IEB terminates communications after transmitting the first byte data.

[Setting condition]

- When the transmit has not been completed although the maximum number of bytes defined by the communications mode have been transmitted.

[Clearing condition]

- When 1 is written.

TXETTME Flag (Transmit Timing Error)

This flag is set to 1 if data is not transmitted with the timing specified by the IEB protocol during data transmission. The IEB sets the TXETTME flag and enters the wait state.

[Setting condition]

- When a timing error occurs during data transmission up to the maximum number of bytes defined by the communications mode.

[Clearing condition]

- When 1 is written.

TXEAL Flag (Arbitration Loss)

The IEB retransmits from the start bit for the number of times specified by setting the RN bits in IEMCR if the arbitration has been lost in master communications. If the arbitration has been lost for the specified number of times, the TXEAL flag is set to enter the wait state. If the arbitration has been won within retransmit for the specified number of times, the TXEAL flag is not set to 1. The TXEAL flag is set only when the arbitration has been lost and the wait state is entered.

[Setting condition]

- When the arbitration has been lost during data transmission and the transmission has been terminated.

[Clearing condition]

- When 1 is written.

TXF Flag (Transmit Normal Completion)

This flag indicates that data for the number of bytes specified by the message length bits has been transmitted with no error.

[Setting condition]

- When data for the number of bytes specified by the message length bits has been transmitted during transfer up to the maximum transfer bytes per frame.

[Clearing condition]

- When 1 is written.

TXS Flag (Transmit Start)

This flag indicates that the IEB starts transmission.

[Setting condition]

- When the arbitration is won and the master address field transmission is completed in master transmission.

[Clearing condition]

- When 1 is written.

38.2.17 IEBus Transmit Interrupt Enable Register (IEIET)

Address(es): 0008 A812h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TXSE	TXFE	—	TXEALE	TXETTMEE	TXEROE	TXEACKE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXEACKE	Acknowledge Interrupt Enable	0: Disables an acknowledge bit (TXEACK) interrupt 1: Enables an acknowledge bit (TXEACK) interrupt	R/W
b1	TXEROE	Transmit-Frame Maximum Transfer Byte Overflow Interrupt Enable	0: Disables a transmit-frame maximum transfer byte overflow (TXERO) interrupt 1: Enables a transmit-frame maximum transfer byte overflow (TXERO) interrupt	R/W
b2	TXETTMEE	Transmit Timing Error Interrupt Enable	0: Disables a transmit timing error (TXETTME) interrupt 1: Enables a transmit timing error (TXETTME) interrupt	R/W
b3	TXEALE	Arbitration Loss Interrupt Enable	0: Disables an arbitration loss (TXEAL) interrupt 1: Enables an arbitration loss (TXEAL) interrupt	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TXFE	Transmit Normal Completion Interrupt Enable	0: Disables a transmit normal completion (TXF) interrupt 1: Enables a transmit normal completion (TXF) interrupt	R/W
b6	TXSE	Transmit Start Interrupt Enable	0: Disables a transmit start (TXS) interrupt 1: Enables a transmit start (TXS) interrupt	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

38.2.18 IEBus Receive Status Register (IERSR)

Address(es): 0008 A814h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXBSY	RXS	RXF	RXEDE	RXEOVE	RXERTME	RXEDLE	RXEPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RXEPE	Parity Error Flag	0: No parity error has occurred 1: A parity error has occurred	R/(W) *1
b1	RXEDLE	Receive-Frame Maximum Transfer Byte Overflow Flag	0: No overrun error has occurred 1: An overrun error has occurred	R/(W) *1
b2	RXERTME	Receive Timing Error Flag	0: No receive timing error has occurred 1: A receive timing error has occurred	R/(W) *1
b3	RXEOVE	Receive Overrun Flag	0: No receive overrun error has occurred 1: A receive overrun error has occurred	R/(W) *1
b4	RXEDE	Broadcast Receive Error Flag	0: No broadcast receive error has occurred 1: A broadcast receive error has occurred	R/(W) *1
b5	RXF	Receive Normal Completion Flag	0: Reception has not been completed successfully 1: Reception has been completed successfully	R/(W) *1
b6	RXS	Receive Start Flag	0: Reception has not started 1: Reception has started	R/(W) *1
b7	RXBSY	Receive Busy Flag	0: Receive data buffer is ready for reception 1: Receive data buffer is busy	R/(W) *1

Note 1. Only 1 can be written to clear the flag.

Each status flag in IERSR corresponds to a bit in the IEIER that enables or disables each interrupt. IERSR is cleared by writing 1 to each bit.

RXEPE Flag (Parity Error)

This flag indicates that a parity error has occurred during data field reception. If a parity error occurs before data field reception, the IEB immediately enters the wait state and the RXEPE flag is not set. If a parity error occurs during data field reception when the maximum number of transfer bytes in one frame have not been received, the RXEPE flag is not set yet. When a parity error occurs, the IEB returns a NAK to the communications destination unit via the acknowledge bit. In this case, the communications destination unit continues retransfer up to the maximum number of transfer bytes in one frame and if the reception has been completed normally with no parity error, the RXEPE flag is not set. If the parity error is still found when the reception is terminated and wait state is entered before receiving data for the number of bytes specified by the message length, the RXEPE flag is set. In broadcast reception, if a parity error occurs during data field reception, the IEB enters the wait state immediately after setting the RXEPE flag. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag is set, the IEB stops communication and enters the wait state. This flag is not set in this case.

[Setting condition]

- When the parity bit of the last receive data of the data field is not correct after the maximum number of transfer bytes have been received.

[Clearing condition]

- When 1 is written.

RXEDLE Flag (Receive-Frame Maximum Transfer Byte Overflow)

This flag indicates that the data reception has not finished within the maximum number of bytes defined by the communications mode because of a parity error or overrun error causing the retransfer of data, or that reception has not been completed because the message length value exceeds*1 the maximum number of transfer bytes in one frame. The IEB sets the RXEDLE flag and enters the wait state. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag is set, the IEB stops communication and enters the wait state. This flag is not set in this case.

Note 1. When the value specified as the message length is greater than the maximum number of transfer bytes, the IEB terminates communications after receiving the first byte data.

[Setting condition]

- When the reception of the value specified as the message length has not been completed within the maximum number of bytes defined by communications mode.

[Clearing condition]

- When 1 is written.

RXERTME Flag (Receive Timing Error)

This flag is set to 1 if data is not received with the timing specified by the IEB protocol during data reception. The IEB sets the RXERTME flag and enters the wait state. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag (RXS) is set, the IEB stops communication and enters the wait state. The RXERTME flag is not set in this case.

[Setting condition]

- When a timing error occurs during data reception of the maximum number of bytes defined by communications mode.

[Clearing condition]

- When 1 is written.

RXEOVF Flag (Receive Overrun)

This flag indicates the overrun during data reception.
The IEB sets the RXEOVF flag when the IEB receives the next byte data while the receive data has not been read (the RXBSY flag is not cleared). In this case, the IEB assumes that an overrun error has occurred and returns a NAK to the communications destination unit. The communications destination unit retransmits data up to the maximum number of transfer bytes. The IEB, however, returns a NAK when the RXBSY flag remains set. If the RXBSY flag is cleared to 0, the IEB returns an ACK and receives the next data. In broadcast reception, if the RXBSY flag is set when data reception starts, the IEB immediately enters the wait state. This flag becomes enabled only after the receive start flag (RXS) is set.

[Setting condition]

- When the next byte data is received while the RXBSY flag is not cleared.

[Clearing condition]

- When 1 is written.

RXEDE Flag (Broadcast Receive Error)

This flag indicates that data could not be received because the receive buffer is not in the receive enabled state (when the IECTR.RE bit is not set to 1 or the IERSR.RXBSY flag is set.) during control field reception in broadcast reception. The RXEDE flag functions when the DEE bit in IECTR is set to 1.

[Setting condition]

- When data could not be received during broadcast reception.

[Clearing condition]

- When 1 is written.

RXF Flag (Receive Normal Completion)

This flag indicates that data for the number of bytes specified by the message length bits has been received successfully.

[Setting condition]

- When data for the number of bytes specified by the message length bits has been received successfully.

[Clearing condition]

- When 1 is written.

RXS Flag (Receive Start)

This flag indicates that the IEB has started reception.

[Setting condition]

- When the data up to message length field has been received correctly from the master unit in slave reception.

[Clearing condition]

- When 1 is written.

RXBSY Flag (Receive Busy)

This flag indicates that the receive data is stored in the receive data buffer (IERB001 to IERB032). Clear this flag after reading out all data. The next receive data cannot be received while the RXBSY flag is set.

[Setting condition]

- When all receive data has been written to the receive data buffer.

[Clearing condition]

- When 1 is written.

38.2.19 IEBus Receive Interrupt Enable Register (IEIER)

Address(es): 0008 A815h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXBSYE	RXSE	RXFE	RXEDEE	RXEOVEE	RXERTMEE	RXEDLEE	RXEPEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RXEPEE	Parity Error Interrupt Enable	0: Disables a parity error (RXEPE) interrupt 1: Enables a parity error (RXEPE) interrupt	R/W
b1	RXEDLEE	Receive-Frame Maximum Transfer Byte Overflow Interrupt Enable	0: Disables a receive-frame maximum transfer byte overflow (RXEDLE) interrupt 1: Enables a receive-frame maximum transfer byte overflow (RXEDLE) interrupt	R/W
b2	RXERTMEE	Receive Timing Error Interrupt Enable	0: Disables a receive timing error (RXERTME) interrupt 1: Enables a receive timing error (RXERTME) interrupt	R/W
b3	RXEOVEE	Receive Overrun Interrupt Enable	0: Disables an overrun control flag (RXEOVE) interrupt 1: Enables an overrun control flag (RXEOVE) interrupt	R/W
b4	RXEDEE	Broadcast Receive Error Interrupt Enable	0: Disables a broadcast receive error (RXEDE) interrupt 1: Enables a broadcast receive error (RXEDE) interrupt	R/W
b5	RXFE	Receive Normal Completion Interrupt Enable	0: Disables a receive normal completion (RXF) interrupt 1: Enables a receive normal completion (RXF) interrupt	R/W
b6	RXSE	Receive Start Interrupt Enable	0: Disables a receive start (RXS) interrupt 1: Enables a receive start (RXS) interrupt	R/W
b7	RXBSYE	Receive Busy Interrupt Enable	0: Disables a receive busy (RXBSY) interrupt 1: Enables a receive busy (RXBSY) interrupt	R/W

38.2.20 IEBus Clock Selection Register (IECKSR)

Address(es): 0008 A818h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLT	FCKS[1:0]		CKS3	SRSTP	CKS[2:0]		
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Input Clock Selection*1	b2 b1 b0 0 0 0: Setting prohibited 0 0 1: 1/2 divided peripheral clock (IECLK) is used (IECLK = 12 MHz, 12.58 MHz) 0 1 0: 1/3 divided peripheral clock (IECLK) is used (IECLK = 18 MHz, 18.87 MHz) 0 1 1: 1/4 divided peripheral clock (IECLK) is used (IECLK = 24 MHz, 25.16 MHz) 1 0 0: 1/5 divided peripheral clock (IECLK) is used (IECLK = 30 MHz, 31.45 MHz) 1 0 1: 1/6 divided peripheral clock (IECLK) is used (IECLK = 36 MHz, 37.74 MHz) 1 1 0: 1/7 divided peripheral clock (IECLK) is used (IECLK = 42 MHz, 44.03 MHz) 1 1 1: Setting prohibited	R/W
b3	SRSTP	Software Reset	0: Software reset is negated 1: Software reset is asserted	R/W
b4	CKS3	Input Clock Selection 3*1,*2	0: Peripheral clock (IECLK) is used 1: Setting prohibited	R/W
b6, b5	FCKS[1:0]	Digital Filter Clock Selection*1	b6 b5 0 0: 1/1 divided peripheral clock (IECLK) is used 0 1: 1/2 divided peripheral clock (IECLK) is used 1 0: 1/3 divided peripheral clock (IECLK) is used 1 1: 1/4 divided peripheral clock (IECLK) is used	R/W
b7	FLT	Digital Filter Enable*1	0: Disables digital filter function 1: Enables digital filter function	R/W

Note 1. Do not change the settings of the FLT, FCKS[1:0], CKS3, and CKS[2:0] bits while IEBus is in transmit/receive operation. IECKSR should be set according to the flowchart for initial setting shown in Figure 38.8.Otherwise, the transmit/receive operation is not guaranteed.

Note 2. The clock supply is stopped except for IECKSR when the MSTPCRC.MSTPC18 bit is set to 1. To use the IEB without specifying low power consumption mode, be sure to clear the MSTPCRC.MSTPC18 bit to 0 to cancel module stop mode.

CKS[2:0] Bits (Input Clock Selection)

These bits select the division ratio for the clock used in the IEB.

When using the digital filter function, the digital filter clock should be set as follows so that the digital filter clock becomes an integral multiple of the clock used in the IEB.

- When the CKS[2:0] bits are set to 001b, the FCKS[1:0] bits should be set to 00b or 01b.
- When the CKS[2:0] bits are set to 010b, the FCKS[1:0] bits should be set to 00b or 10b.
- When the CKS[2:0] bits are set to 011b, the FCKS[1:0] bits should be set to 00b, 01b, or 11b.
- When the CKS[2:0] bits are set to 100b, the FCKS[1:0] bits should be set to 00b.
- When the CKS[2:0] bits are set to 101b, the FCKS[1:0] bits should be set to 00b, 01b, or 10b.
- When the CKS[2:0] bits are set to 110b, the FCKS[1:0] bits should be set to 00b.
- When the CKS[2:0] bits are set to 111b, the FCKS[1:0] bits should be set to 00b, 01b, or 11b.

SRSTP Bit (Software Reset)

When a hardware reset is executed, it sets the software reset and then all registers except the IECKSR register are reset. After the hardware reset, negate the software reset.

Since registers except the IECKSR register have been reset while the software reset is being asserted, accessing to the registers except the IECKSR register is prohibited.

In addition, if bits FLT, FCKS[1:0], CKS3, and CKS[2:0] are modified without following the flowchart for initial setting shown in Figure 38.8, assert or negate the software reset to prevent a malfunction.

38.2.21 CIEBus Transmit Data Buffer Registers 001 to 032 (IETB001 to IETB032)

Address(es): 0008 A900h to 0008 A91Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TBn[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

n = 001 to 032

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TBn[7:0]	IEBus Transmit Data Buffer	Data to be transmitted in the data field during master transmission is written to TB001 to TB032. Data is written starting with TB001 for the start 1-byte data, followed by TB002 and TB003 and so on according to the transmission order, and TB032 stores the last data of 32-byte transmission.	W*1

Note 1. Since writing to these bits during master or slave transmission (IEFLG.MRQ bit is 1, or IEFLG.SRQ is 1) is disabled, accessing to these bits is prohibited.

IETB001 to IETB032 works as a 32-byte (8 × 32) buffer to which data to be transmitted during master transmission is written.

38.2.22 IEBus Receive Data Buffer Registers 001 to 032 (IERB001 to IERB032)

Address(es): 0008 AA00h to 0008 AA1Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	RBn[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

n = 001 to 032

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RBn[7:0]	IEBus Receive Data Buffer	Data in RB001 to RB032 can be read when the RXBSY bit in the IEBus receive status register (IERSR) is set to 1. Data read from RB001 to RB032 is the field data during slave reception. Receive data is written starting with RB001 for the start 1-byte data, followed by RB002 and RB003 and so on, and RB032 stores the last data of 32-byte reception.	R*1

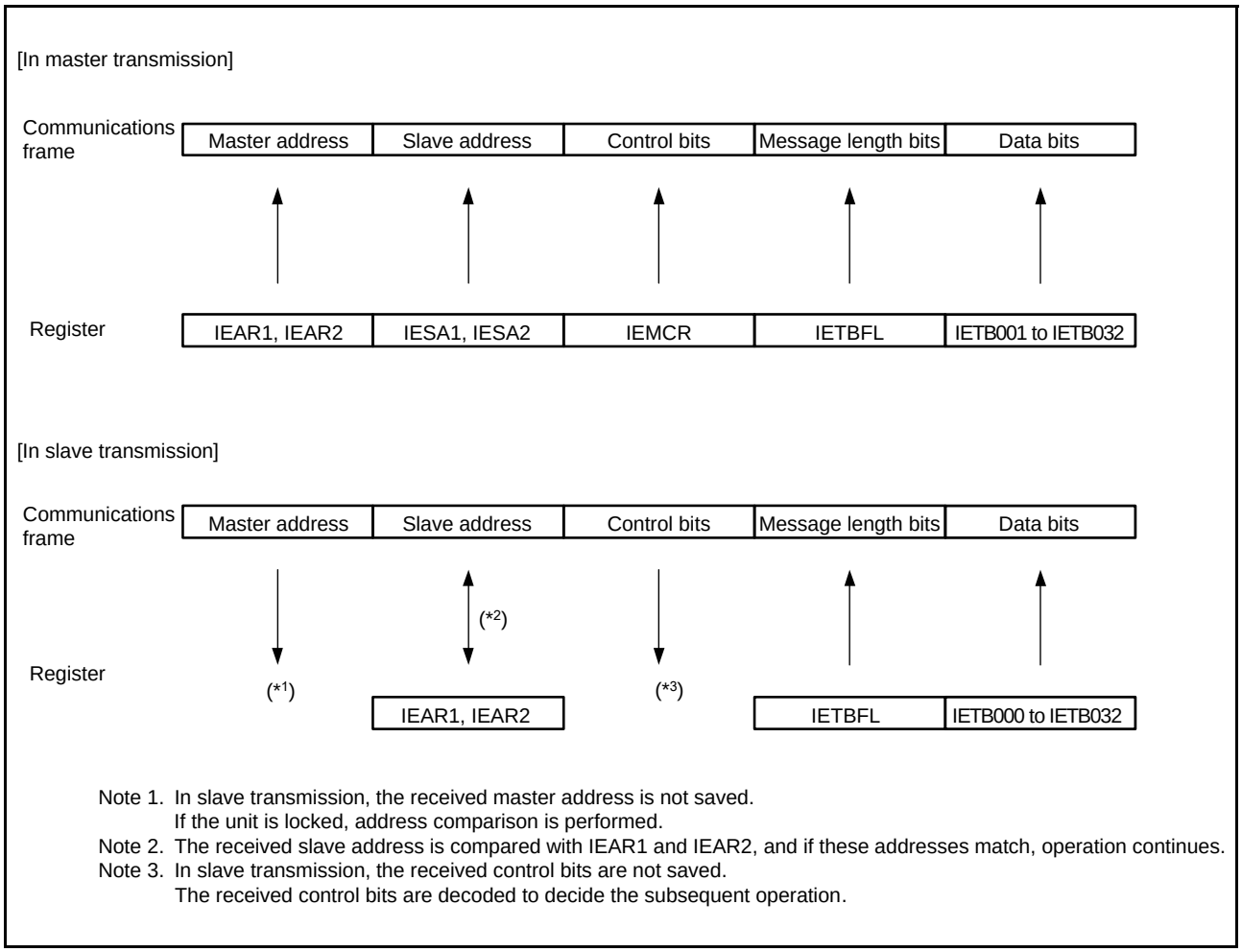
Note 1. Since reading these bits during master reception (after master communications request is issued while IEMCR.CTL[3] is 0, IEFLG.MRQ is 1 and IERSR.RXBSY is 0) or slave reception (IEFLG.SRE is 1 and IERSR.RXBSY is 0) is disabled, accessing to these bits is prohibited. (Read value is undefined.)

IERB001 to IERB032 work as a 32-byte (8 × 32) buffer in which data received during slave reception is stored.

38.3 Data Format

38.3.1 Transmission Format

Figure 38.6 shows the relationship between the transfer format and each register during the IEBus data transmission.



Note 1. In slave transmission, the received master address is not saved.
If the unit is locked, address comparison is performed.
Note 2. The received slave address is compared with IEAR1 and IEAR2, and if these addresses match, operation continues.
Note 3. In slave transmission, the received control bits are not saved.
The received control bits are decoded to decide the subsequent operation.

Figure 38.6 Relationship between Transfer Format and Each Register during IEBus Data Transmission

38.3.2 Reception Format

Figure 38.7 shows the relationship between the transfer format and each register during the IEBus data reception.

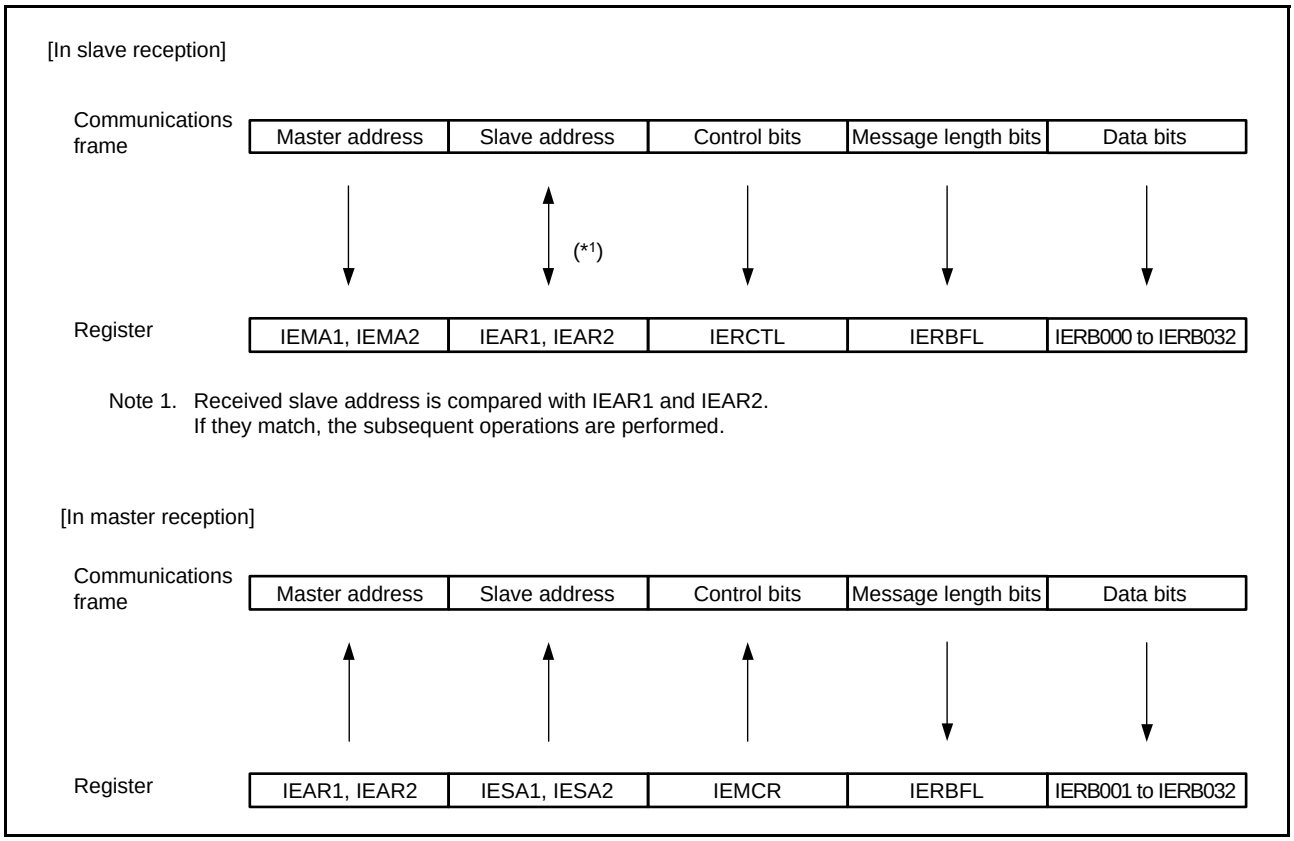
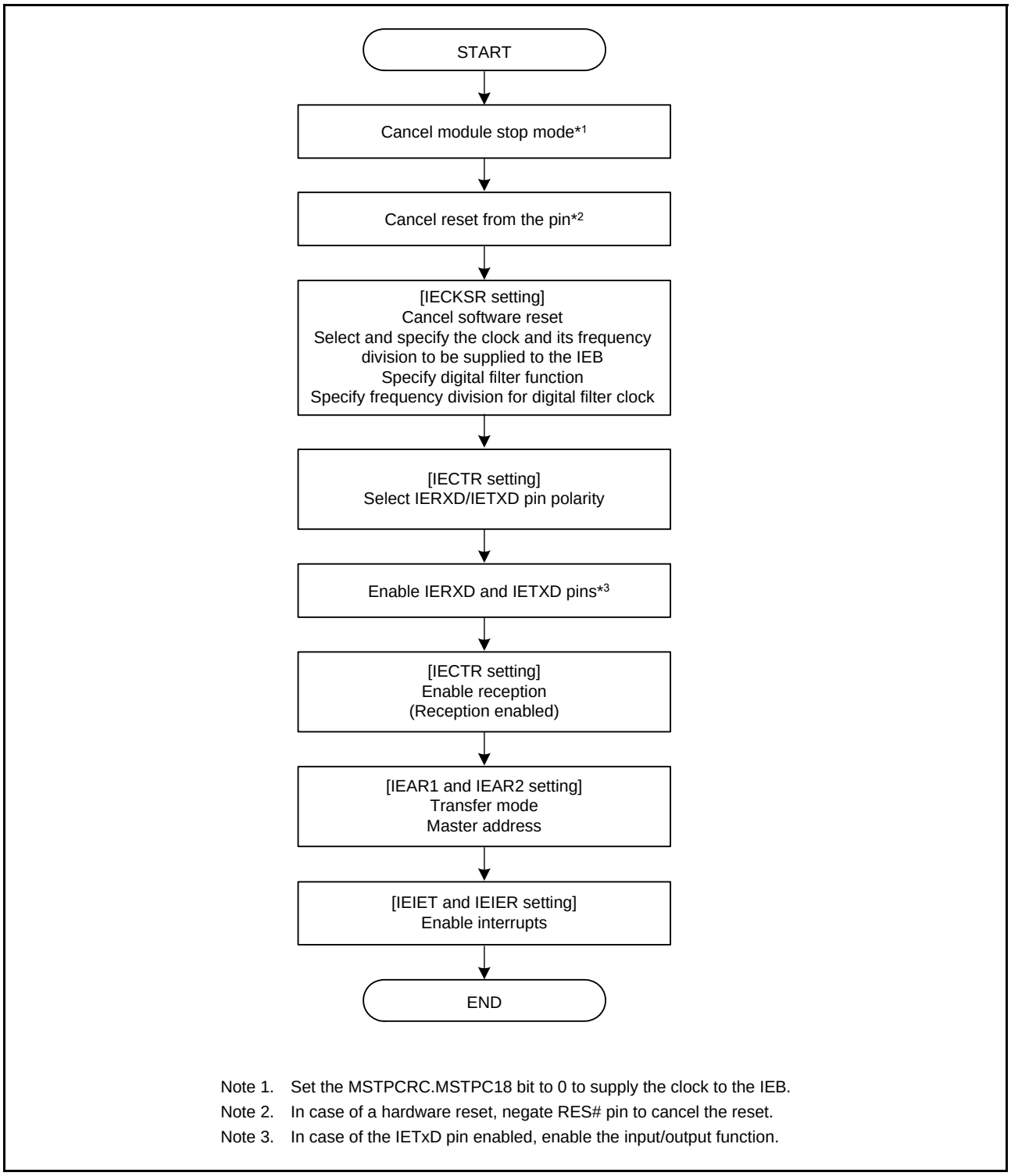


Figure 38.7 Relationship between Transfer Format and Each Register during IEBus Data Reception

38.4 Control Flows

38.4.1 Initial Setting

Figure 38.8 shows the flowchart for the initial setting.



Note 1. Set the MSTPCRC.MSTPC18 bit to 0 to supply the clock to the IEB.
Note 2. In case of a hardware reset, negate RES# pin to cancel the reset.
Note 3. In case of the IETxD pin enabled, enable the input/output function.

Figure 38.8 Flowchart for Initial Setting

38.4.2 Master Transmission

Figure 38.9 shows the flowchart for master transmission.

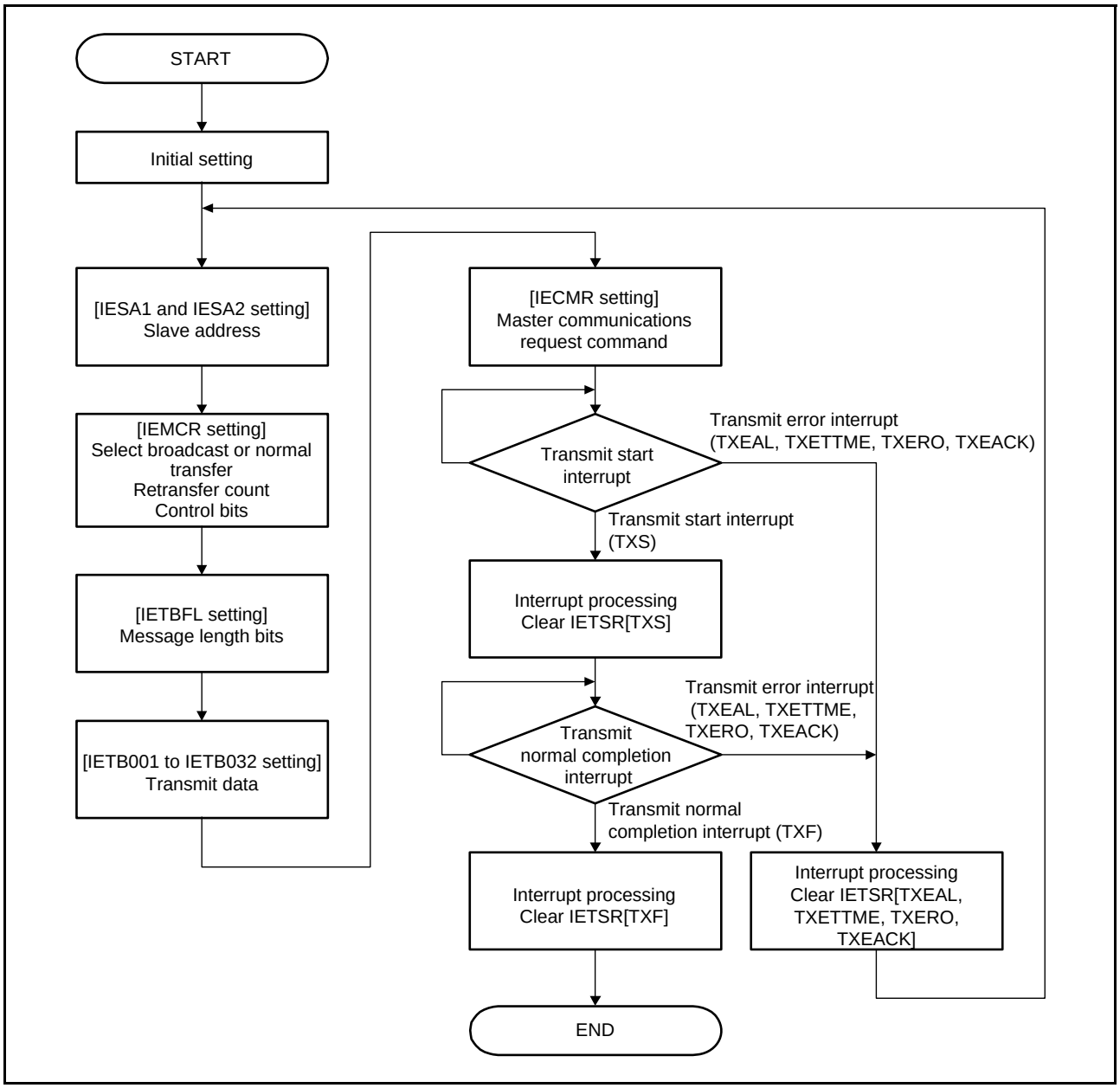


Figure 38.9 Flowchart for Master Transmission

38.4.3 Slave Reception

Figure 38.10 shows the flowchart for slave reception.

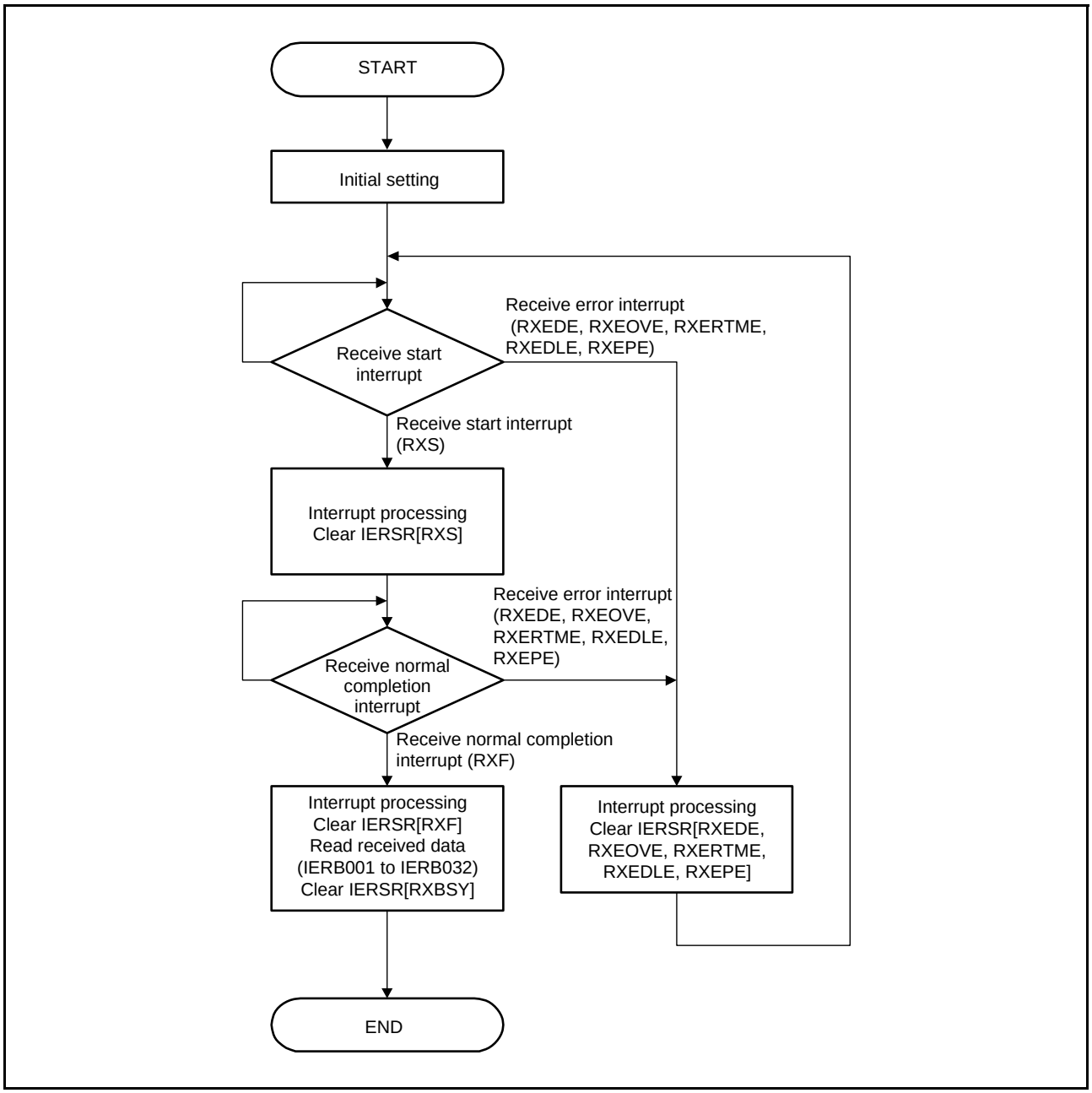


Figure 38.10 Flowchart for Slave Reception

38.4.4 Master Reception

Figure 38.11 shows the flowchart for master reception.

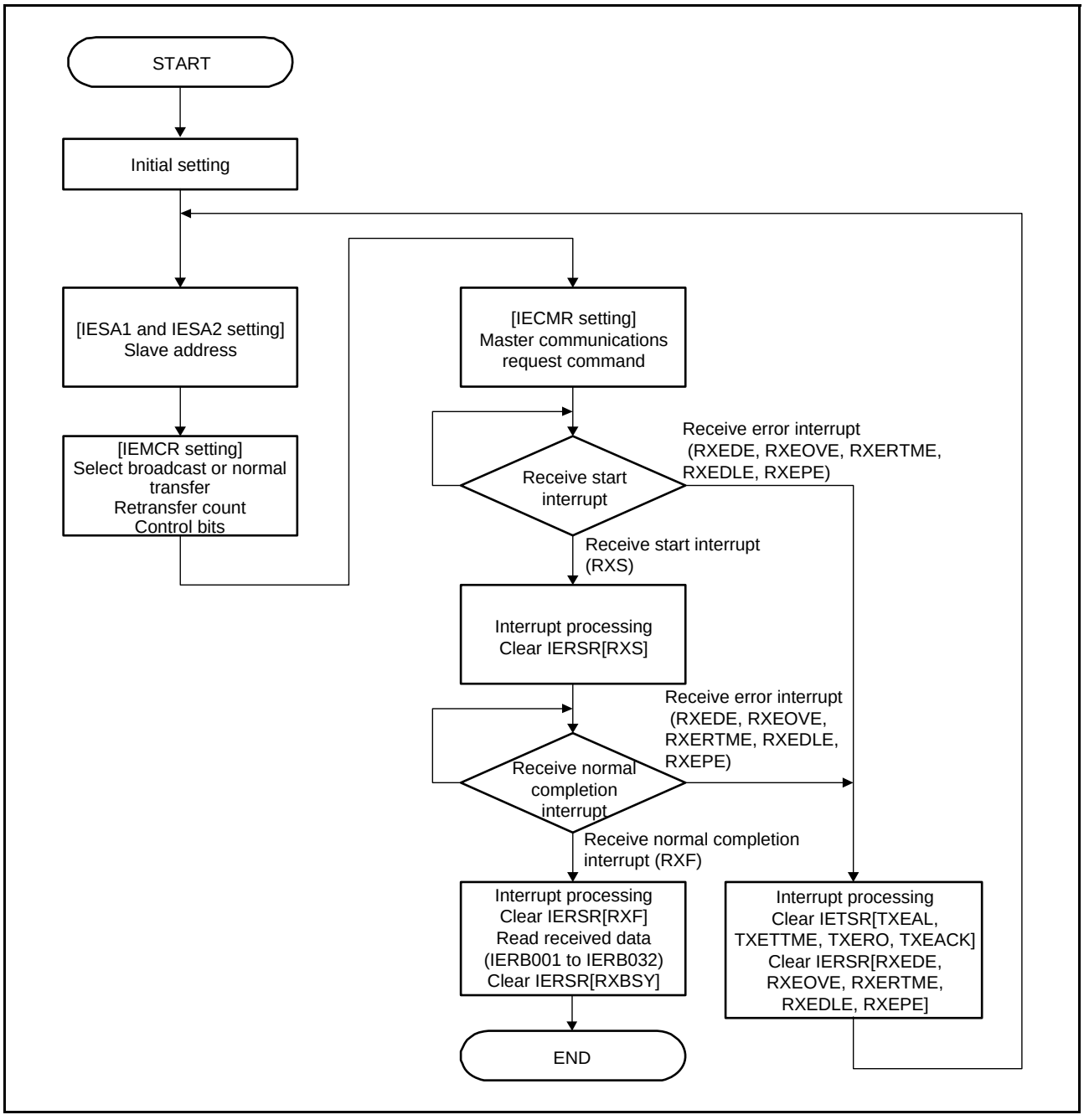


Figure 38.11 Flowchart for Master Reception

38.4.5 Slave Transmission

Figure 38.12 shows the flowchart for slave transmission.

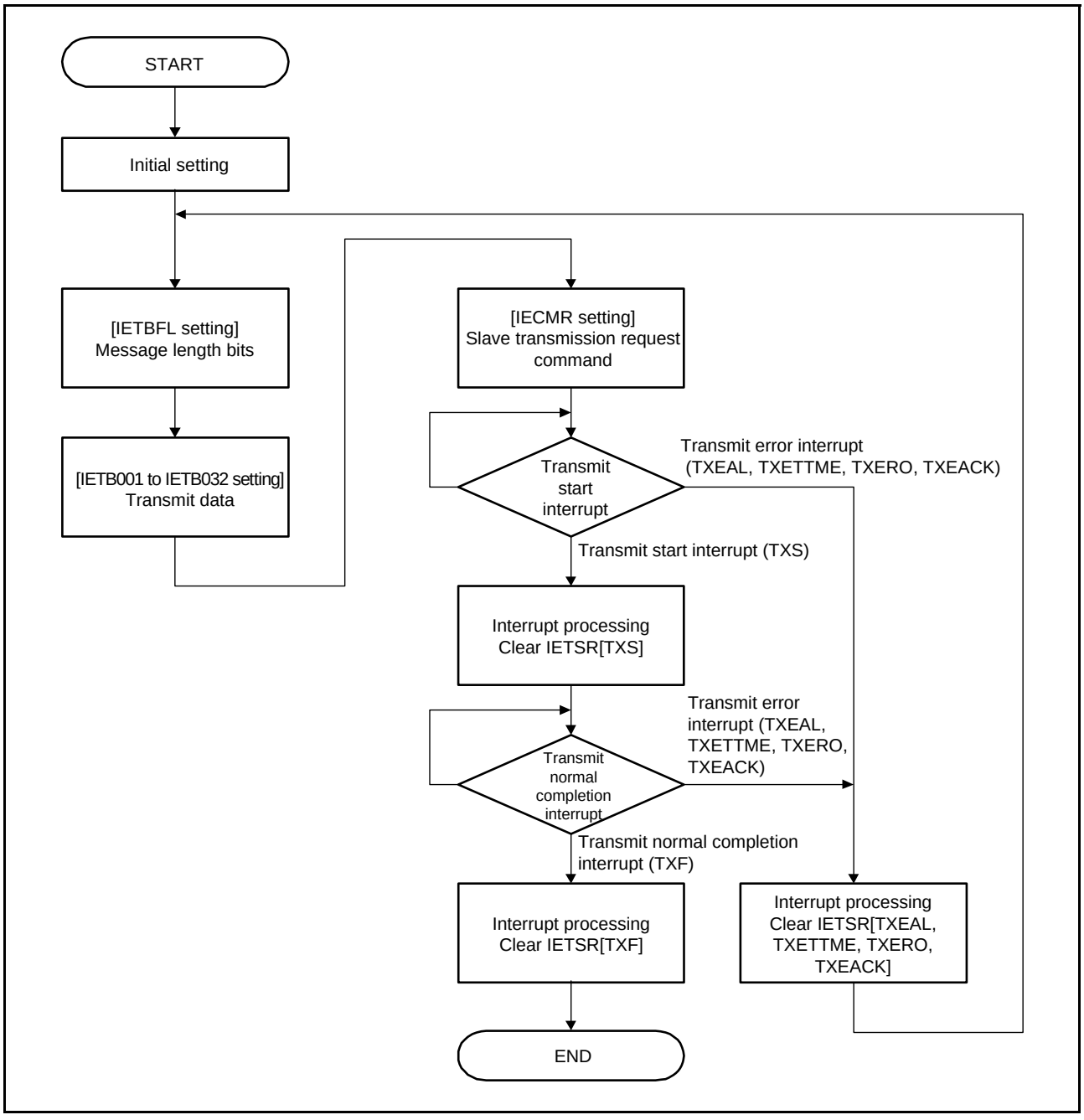


Figure 38.12 Flowchart for Slave Transmission

38.5 Operation Timing

38.5.1 Master Transmission

Figure 38.13 shows the timing for master transmission.

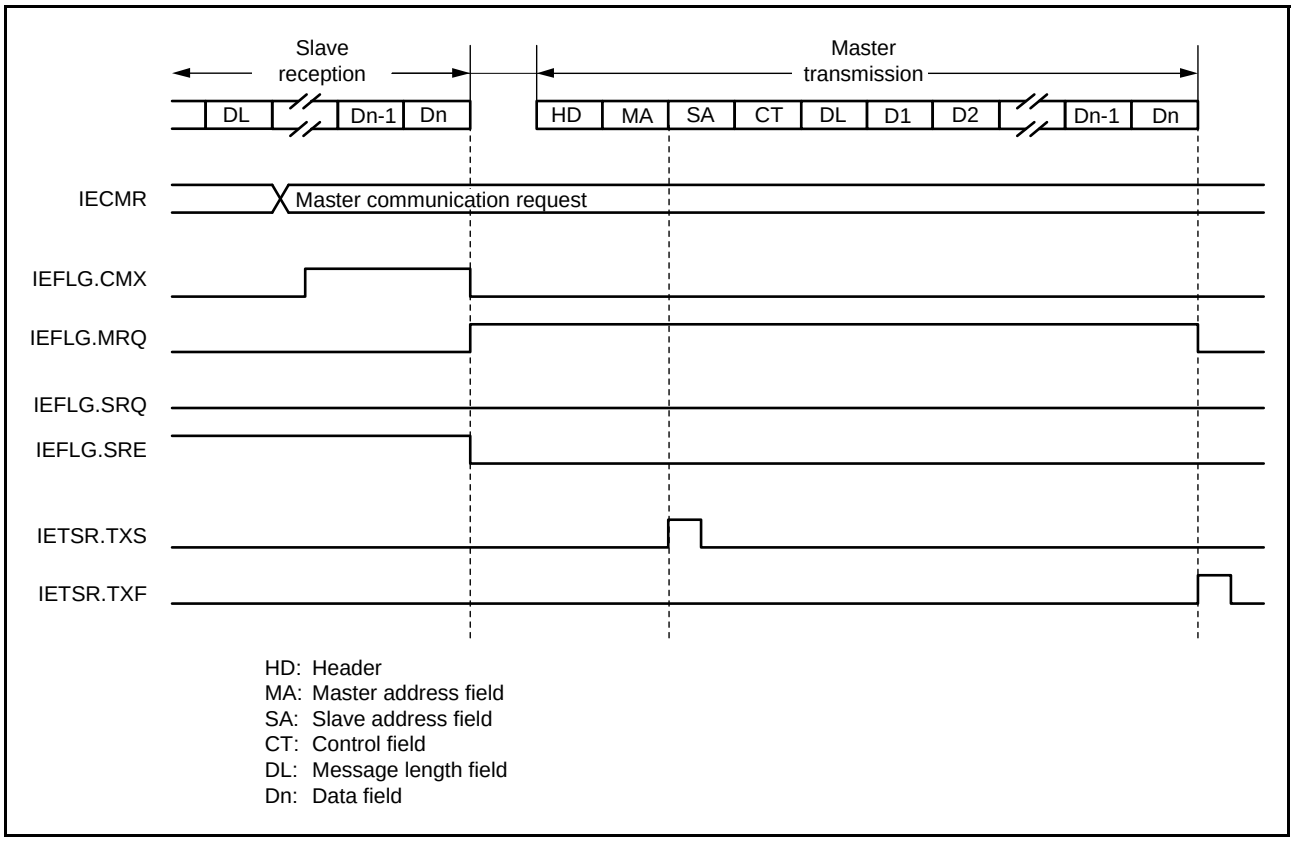


Figure 38.13 Master Transmission Timing

38.5.2 Slave Reception

Figure 38.14 shows the timing for slave reception.

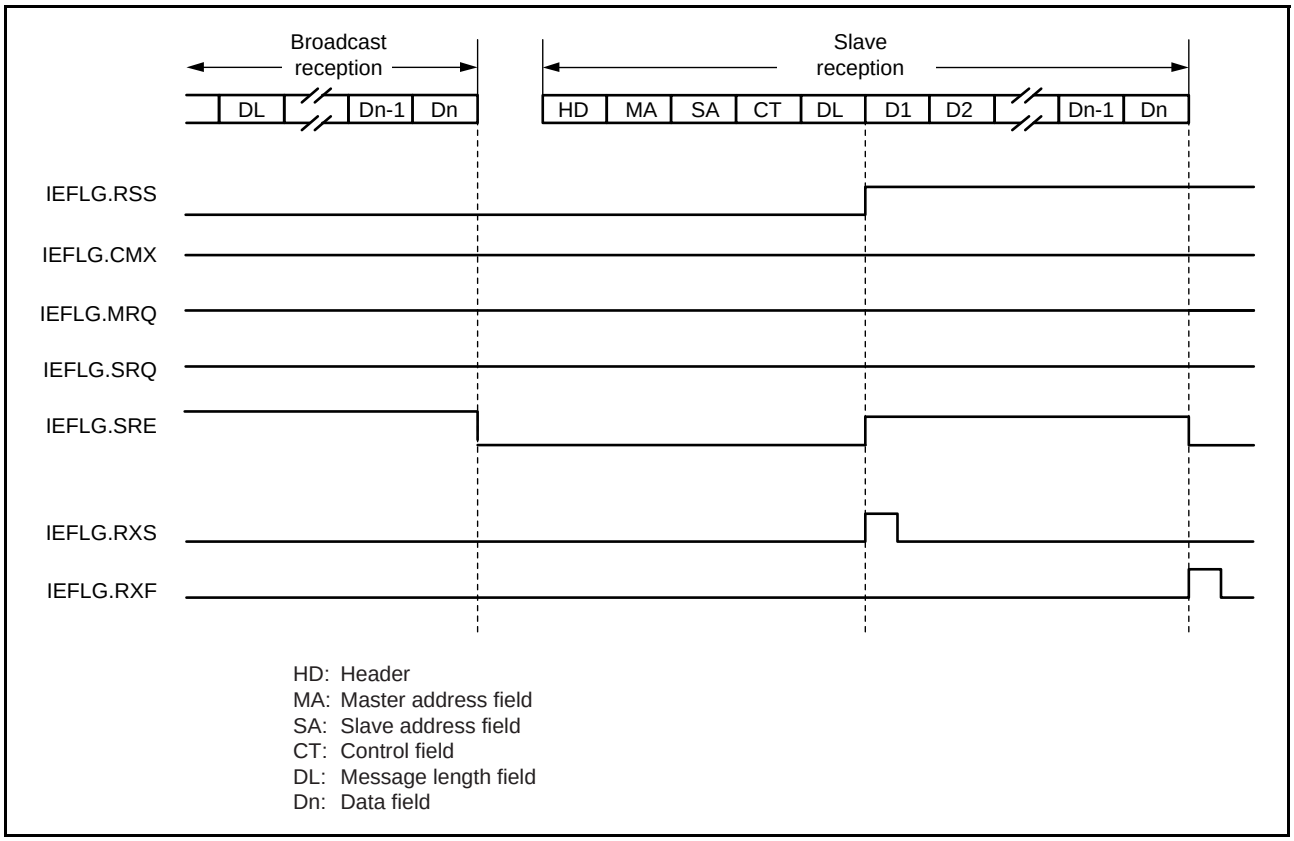


Figure 38.14 Slave Reception Timing

38.5.3 Master Reception

Figure 38.15 shows the timing for master reception.

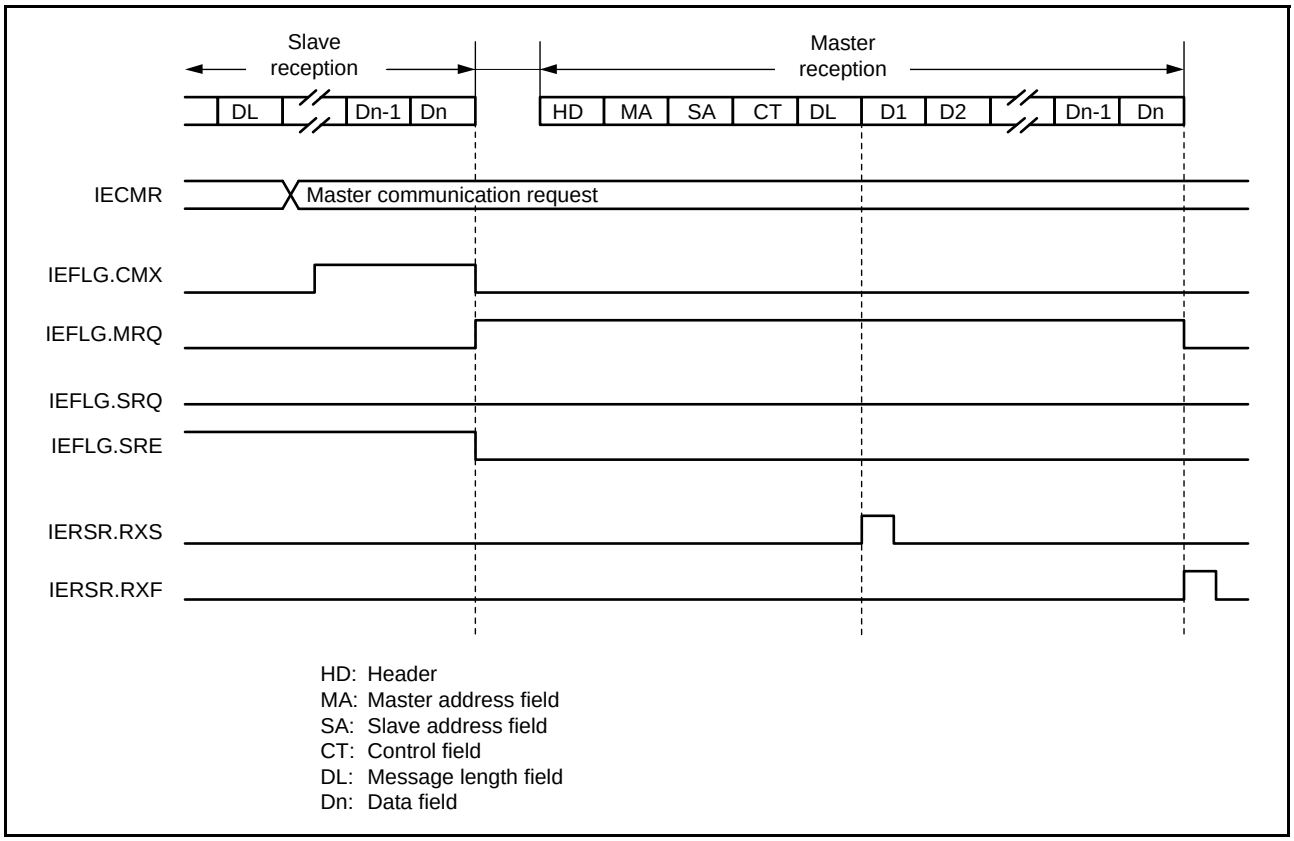


Figure 38.15 Master Reception Timing

38.5.4 Slave Transmission

Figure 38.16 shows the timing for slave transmission.

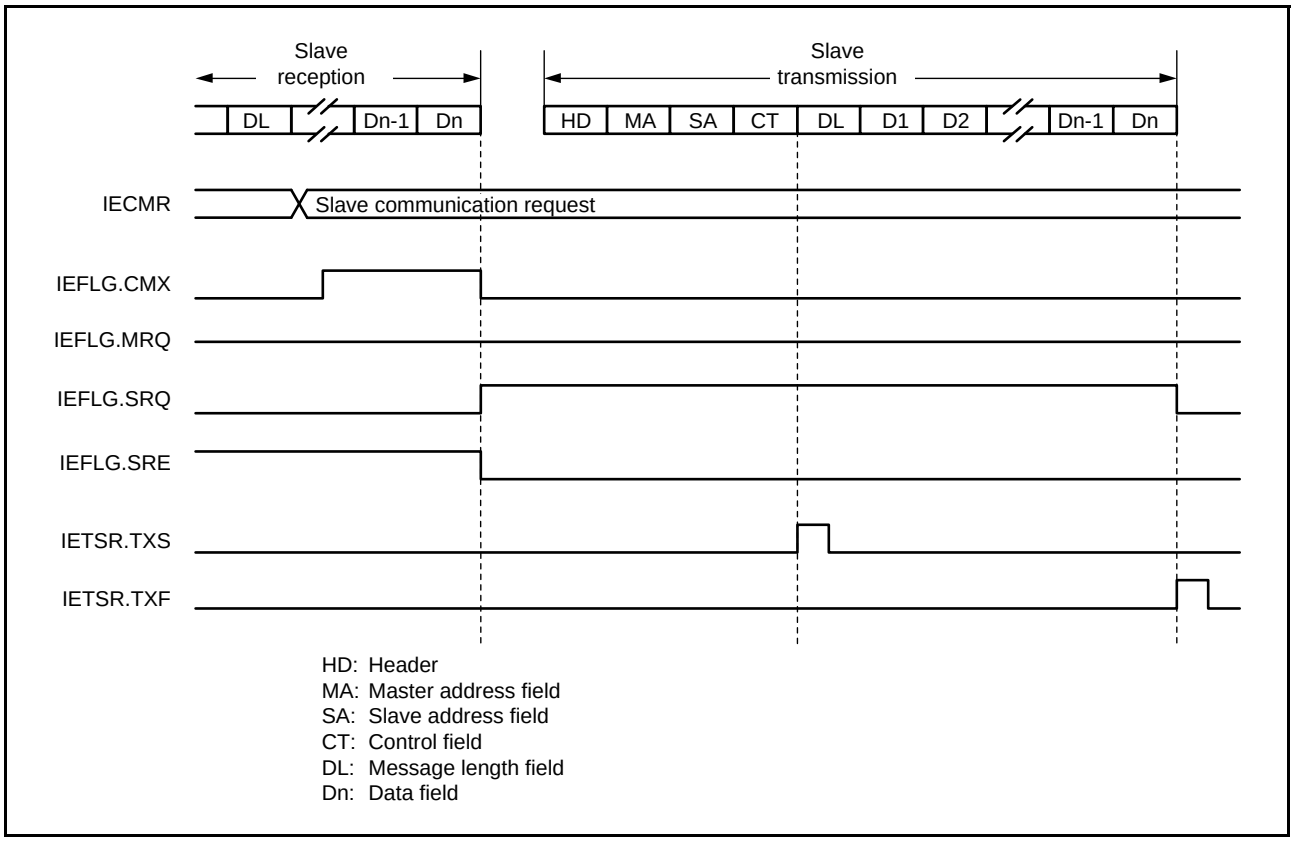


Figure 38.16 Slave Transmission Timing

38.6 Interrupt Sources

The IEB interrupt sources include the following: transmit start (TXS), transmit normal completion (TXF), arbitration loss (TXEAL), transmit timing error (TXETTME), transmit-frame maximum transfer byte overflow (TXERO), acknowledge bit (TXEACK), receive busy (RXBSY), receive start (RXS), receive normal completion (RXF), broadcast receive error (RXEDE), receive overrun flag (RXEOVE), receive timing error (RXERTME), receive-frame maximum transfer byte overflow (RXEDLE), and parity error (RXEPE).
Each source has a corresponding bit in the IEBus transmit interrupt enable register (IEIET) or the IEBus receive interrupt enable register (IEIER), which enables or disables the interrupt. Each source also has a corresponding status flag in the IEBus transmit status register (IETSR) or IEBus receive status register (IERSR). Reading the status flags allows determination of the interrupt sources.
Figure 38.17 shows the relations between the IEB interrupt sources.

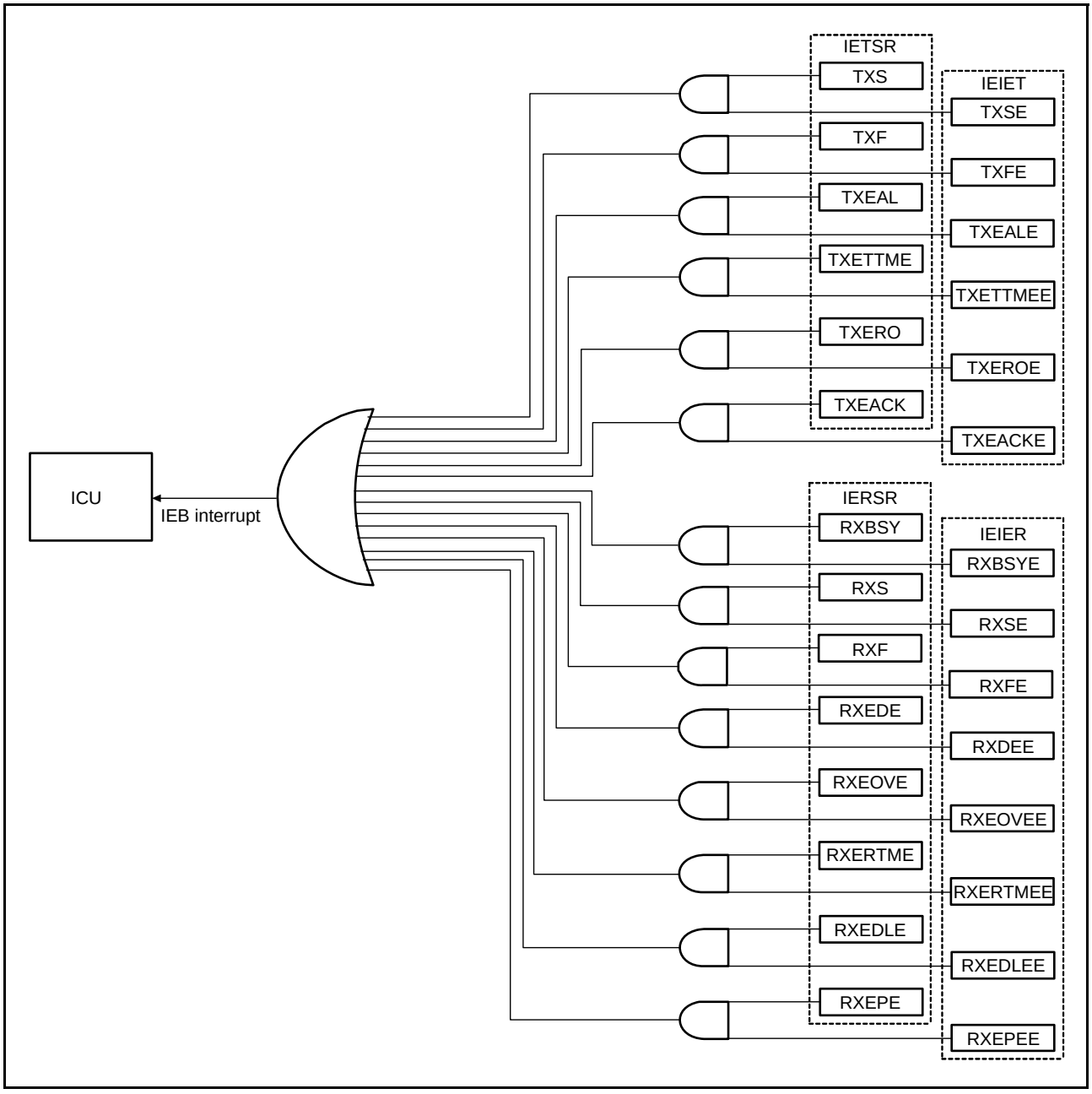


Figure 38.17 Relations between IEB Interrupt Sources

38.7 Usage Notes

38.7.1 Notes on Operation when Transfer Is Not Completed within Maximum Number of Transfer Bytes

(1) Data Transmission

When data has been transmitted up to the maximum number of bytes defined by the communications mode but transmission has not been completed because a NAK has been received, from the receive unit during data transmission the IEB sets the error flag in IESTR and enters a wait state.
When a timing error has occurred during transmission up to the number of bytes specified as the message length, the IEB sets the IETSR.TXETTME flag and enters a wait state.
When data has been transferred for the number of bytes specified as the message length within the maximum number of bytes defined by the communications mode, if an ACK has been received, the IEB does not set the IETSR.TXERO or IETSR.TXEACK flag but sets the IETSR.TXF flag. If a NAK has been received, the IEB sets the IETSR.TXERO and IETSR.TXEACK flags.
When data has been transferred up to the maximum number of bytes defined by the communications mode but the transferred bytes are less than the message length, if an ACK has been received, the IEB sets the IETSR.TXERO flag. If a NAK has been received, the IEB sets the IETSR.TXERO and IETSR.TXEACK flags.
Figure 38.18 shows the timing of operations when transmission has not been completed within the maximum number of bytes.

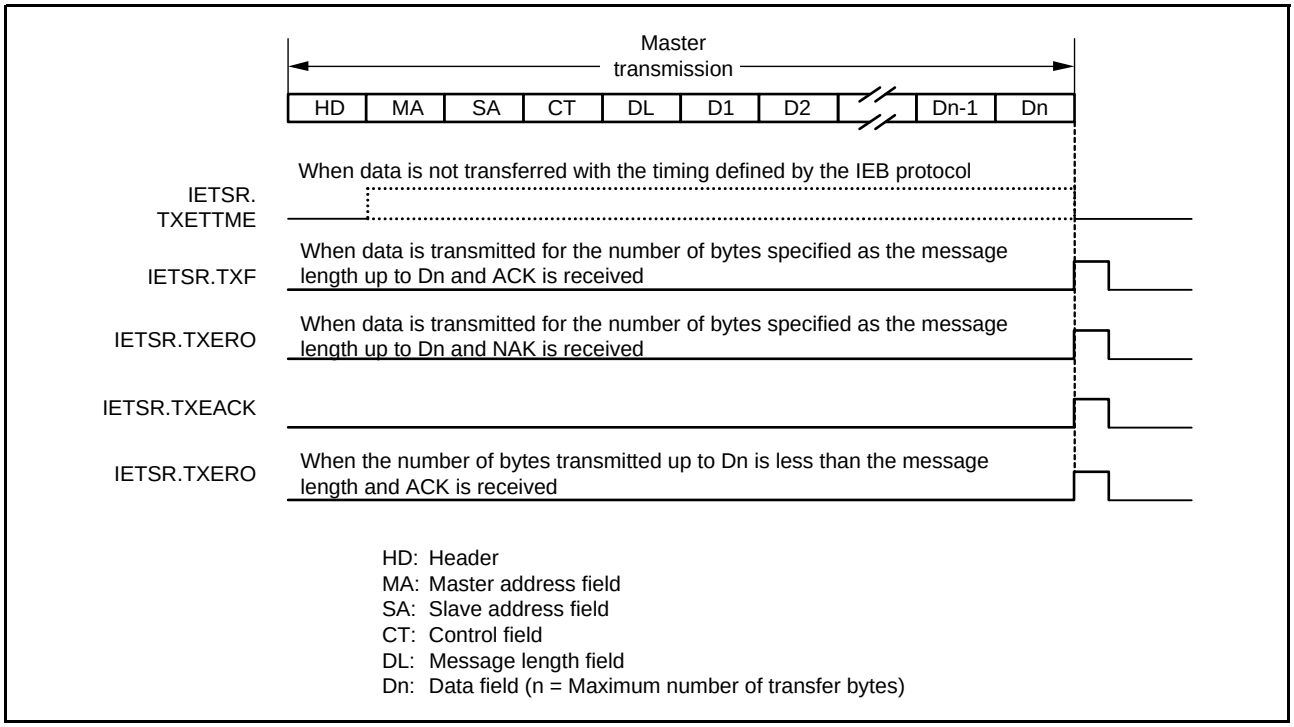


Figure 38.18 Timing of Operations when Transmission is Not Completed within Maximum Number of Transfer Bytes

(2) Data Reception

When data reception has not been completed within the maximum number of bytes defined by the communications mode because of a parity error or overrun error causing the retransfer of data, the IEB sets the error flag in IERSR and enters a wait state.

When a timing error has occurred during reception up to the number of bytes specified as the message length, the IEB sets the IERSR.RXERTME flag and enters a wait state.

When data has been received for the number of bytes specified as the message length within the maximum number of bytes defined by the communications mode, if no parity error has been detected, the IEB does not set the IERSR.RXEDLE or IERSR.RXEPE flag but sets the IERSR.RXF flag. If a parity error has occurred, the IEB sets the IERSR.RXEDLE and IERSR.RXEPE flags.

When data has been received up to the maximum number of bytes defined by the communications mode but the transferred bytes are less than the message length, if no parity error has been detected, the IEB sets the IERSR.RXEDLE flag. If a parity error has occurred, the IEB sets the IERSR.RXEDLE and IERSR.RXEPE flags.

When the IERSR.RXEDLE flag is set, discard the reception data because data has not been normally written to the receive buffer.

Figure 38.19 shows the timing of operations when the maximum number of transfer bytes is reached but reception has not been completed.

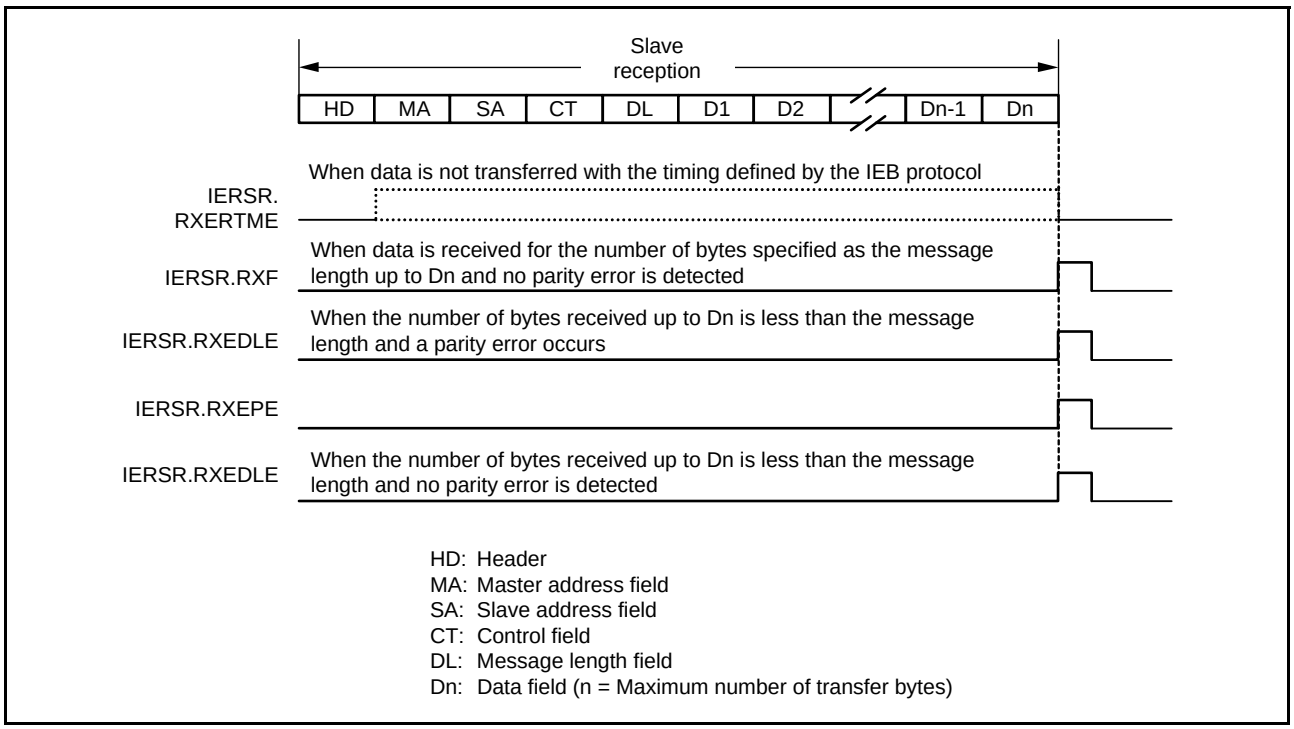


Figure 38.19 Timing of Operations when Reception is Not Completed within Maximum Number of Transfer Bytes

38.7.2 Notes on Operation when Message Length Is Greater than Maximum Number of Transfer Bytes

In the RX63N/RX631 Group, the value specified in the message length bits should not be greater than the maximum number of transfer bytes in one frame.
The following describes operations when a greater value is specified by mistake.

(1) Data Transmission

When the value specified in the IEBus transmit message length register (IETBFL) is greater than the maximum number of transfer bytes in data transmission, the IEB terminates transfer after transmission of the first byte data and sets the transmit-frame maximum transfer byte overflow (TXERO) flag in the IEBus transmit status register (IETSR).
Figure 38.20 shows the timing of operations when the transmit message length is greater than the maximum number of transfer bytes.

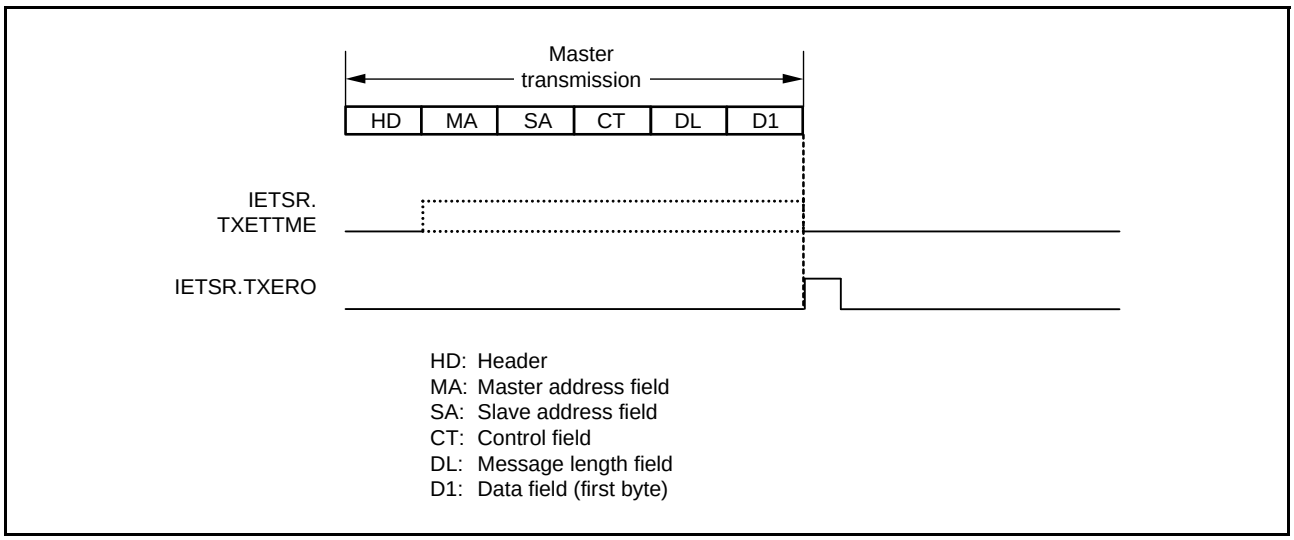


Figure 38.20 Timing of Operations when Transmit Message Length is Greater than Maximum Number of Transfer Bytes

(2) Data Reception

When the value specified in the message length field to be received is greater than the maximum number of transfer bytes in data reception, the IEB terminates transfer after reception of the first byte data and sets the receive-frame maximum transfer byte overflow (RXEDLE) flag in the IEBus receive status register (IERSR).
When the IERSR.RXEDLE flag is set, discard the reception data because data has not been normally written to the receive buffer.
Figure 38.21 shows the timing of operations when the value in the receive message length field is greater than the maximum number of transfer bytes.

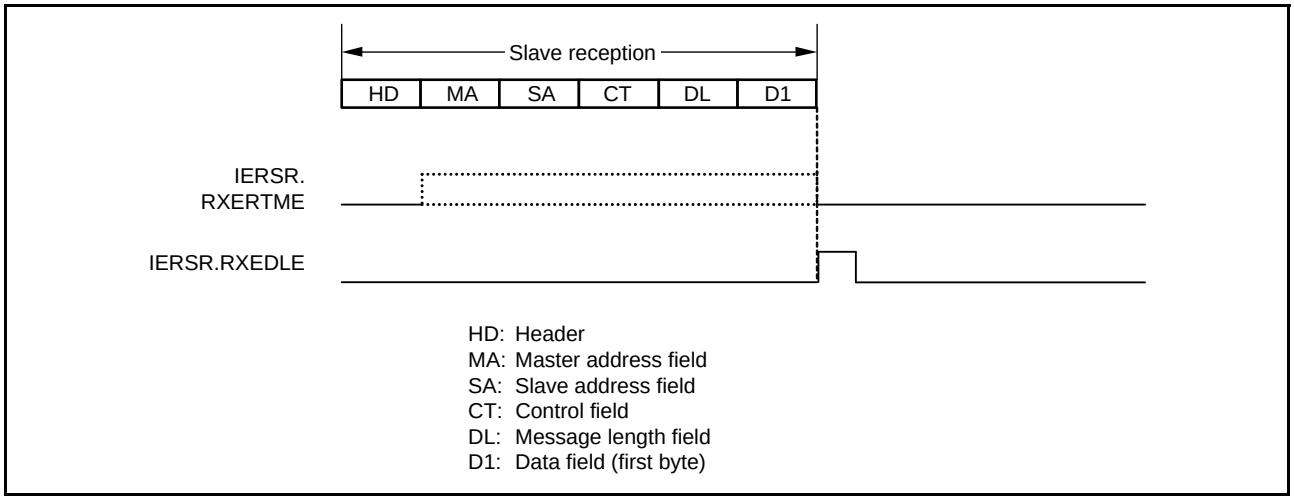


Figure 38.21 Timing of Operations when Value in Receive Message Length Field is Greater than Maximum Number of Transfer Bytes

38.7.3 Setting for the Module-Stop State

Module-stop control register C (MSTPCRC) can be used to enable or disable operation of the IEB. The IEB 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.

39. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes of data blocks.

39.1 Overview

Table 39.1 lists the specifications of the CRC calculator, and Figure 39.1 shows a block diagram of the CRC calculator.

Table 39.1 Specifications of CRC

Item	Description
Data for CRC calculation*1	CRC code generated for any desired data in 8n-bit units (where n is a whole number)
Data block size	8 bits
CRC processor unit	Operation executed on eight bits in parallel
CRC generating polynomial	One of three generating polynomials 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	CRC code generation for LSB-first or MSB-first communication selectable
Low-power consumption function	Module stop state can be set

Note 1. The circuit does not have functionality to divide data for calculation into a data-block size. Write data in 8-bit units.

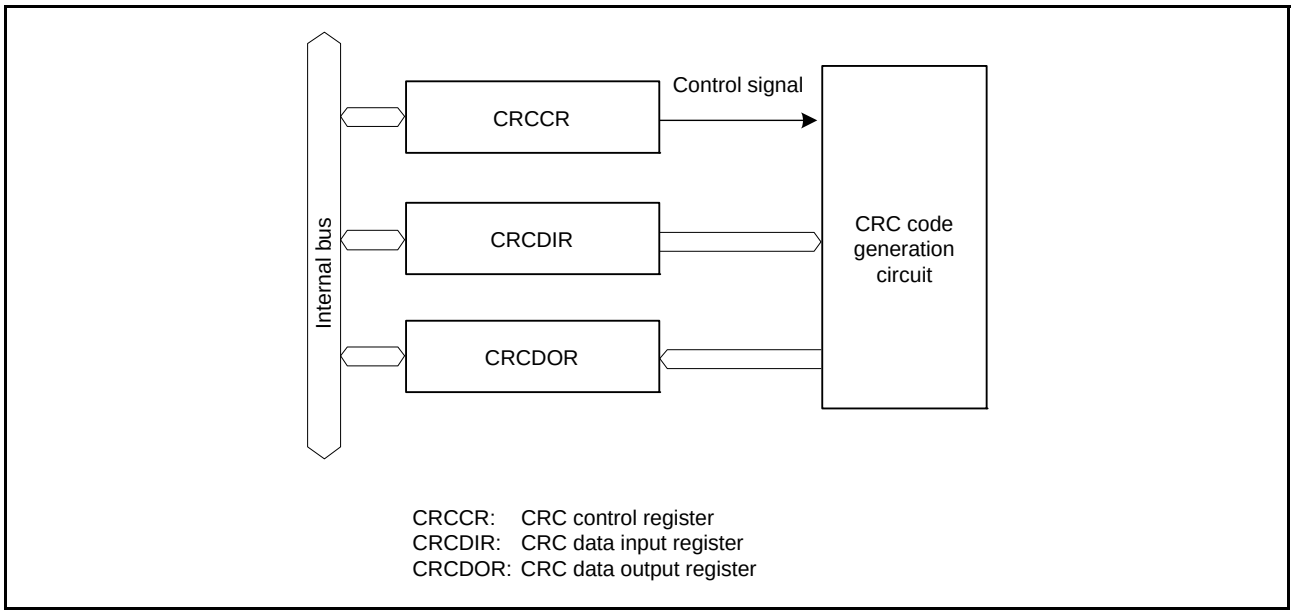


Figure 39.1 Block Diagram of CRC Calculator

39.2 Register Descriptions

39.2.1 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: $X^8 + X^2 + X + 1$ 1 0: $X^{16} + X^{15} + X^2 + 1$ 1 1: $X^{16} + X^{12} + X^5 + 1$	R/W
b2	LMS	CRC Calculation Switching	0: Performs CRC operation for LSB-first communication. The lower-order byte (bits 7 to 0) is the first to be transmitted when the contents of the CRCDOR (CRC code) are divided into bytes. 1: Performs CRC operation for MSB-first communication. The higher-order byte (bits 15 to 8) is first to be transmitted when the contents of the CRCDOR (CRC code) are divided into bytes.	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: Clear the CRCDOR register*1 This bit is read as 0.	W

Note 1. Only 1 can be written.

GPS[1:0] Bits (CRC Generating Polynomial Switching)

These bits select the CRC code generating polynomial.

LMS Bit (CRC Calculation Switching)

Selects LSB-first or MSB-first communication for the CRC code generation.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is cleared to 0000h.
This bit is read as 0. Only 1 can be written.

39.2.2 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 an 8-bit readable/writable register, to which the bytes to be CRC-operated are written.

39.2.3 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 16-bit readable/writable register that contains the result of CRC calculation.
In general, the value will be 0 if there is no CRC error when the calculated CRC code matches the CRC code that continues on, for verification, from the transferred data.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the lower-order byte (b7 to b0). The higher-order byte (b15 to b8) is not updated.

39.3 Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.
The following figures show examples in which the CRCCR.GPS[1:0] bits are set to 11b so the CRC code is calculated by using a 16-bit CRC (with the polynomial $X^{16} + X^{12} + X^5 + 1$), and the CRC code is calculated for the value “F0h”.
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 CRCDOR.

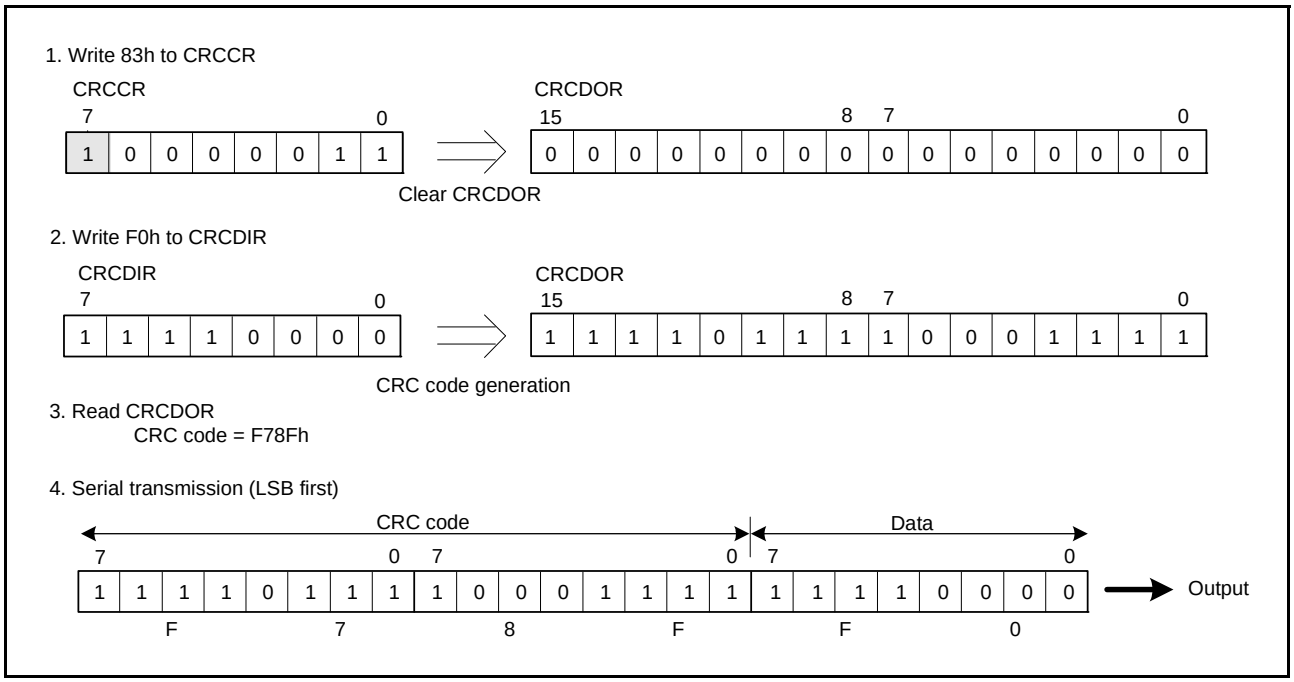


Figure 39.2 LSB-First Data Transmission

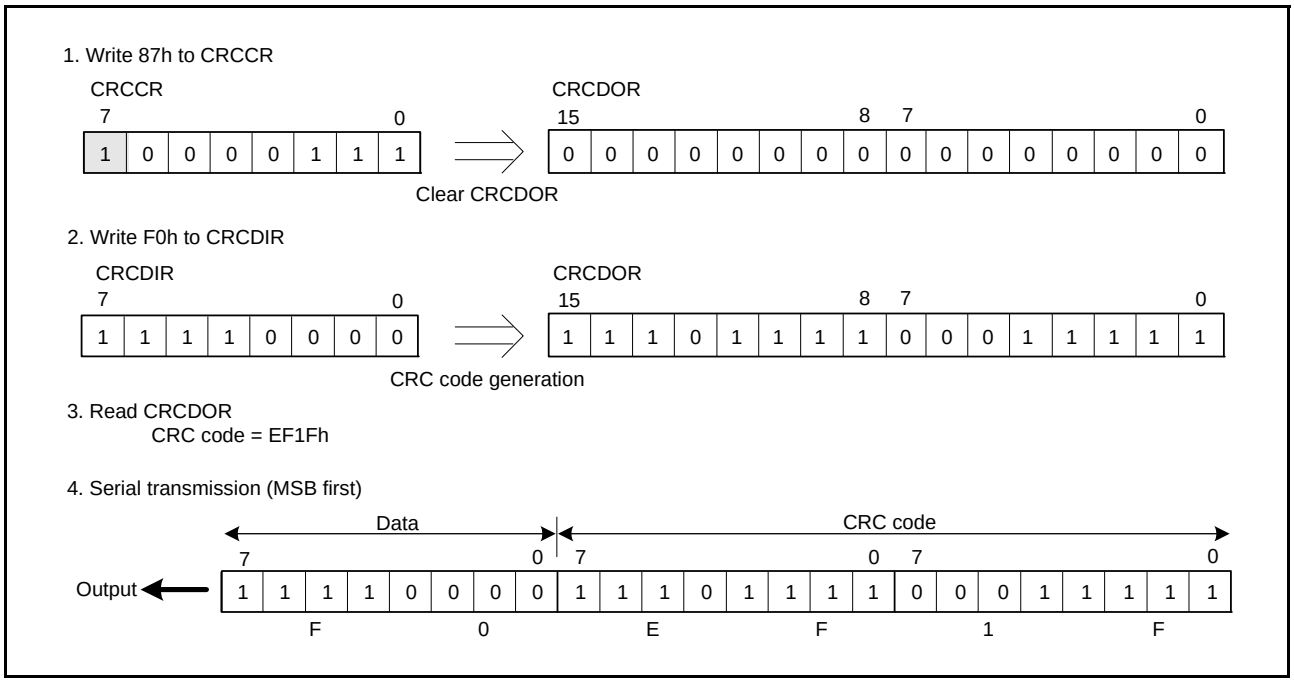


Figure 39.3 MSB-First Data Transmission

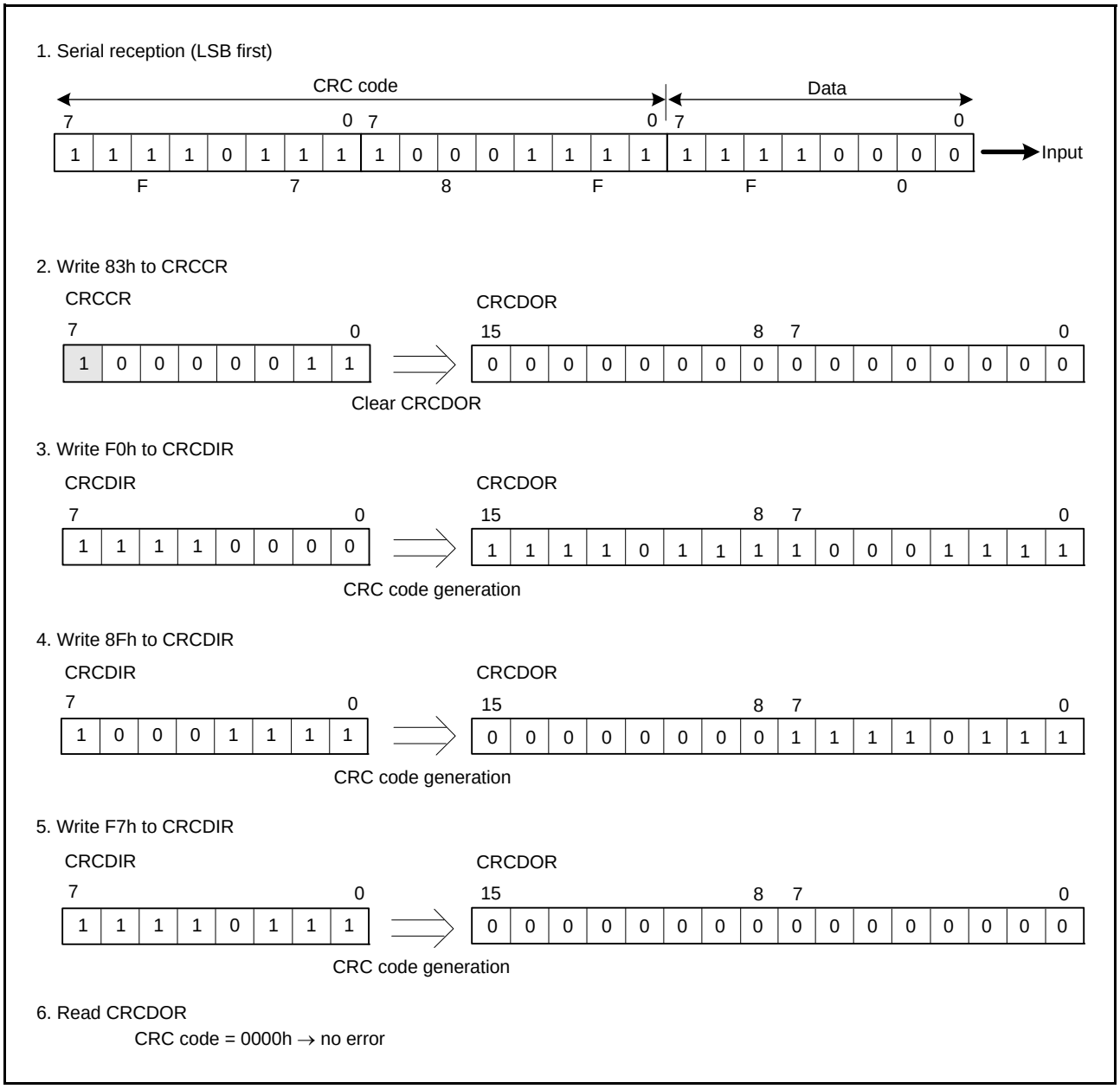


Figure 39.4 LSB-First Data Reception

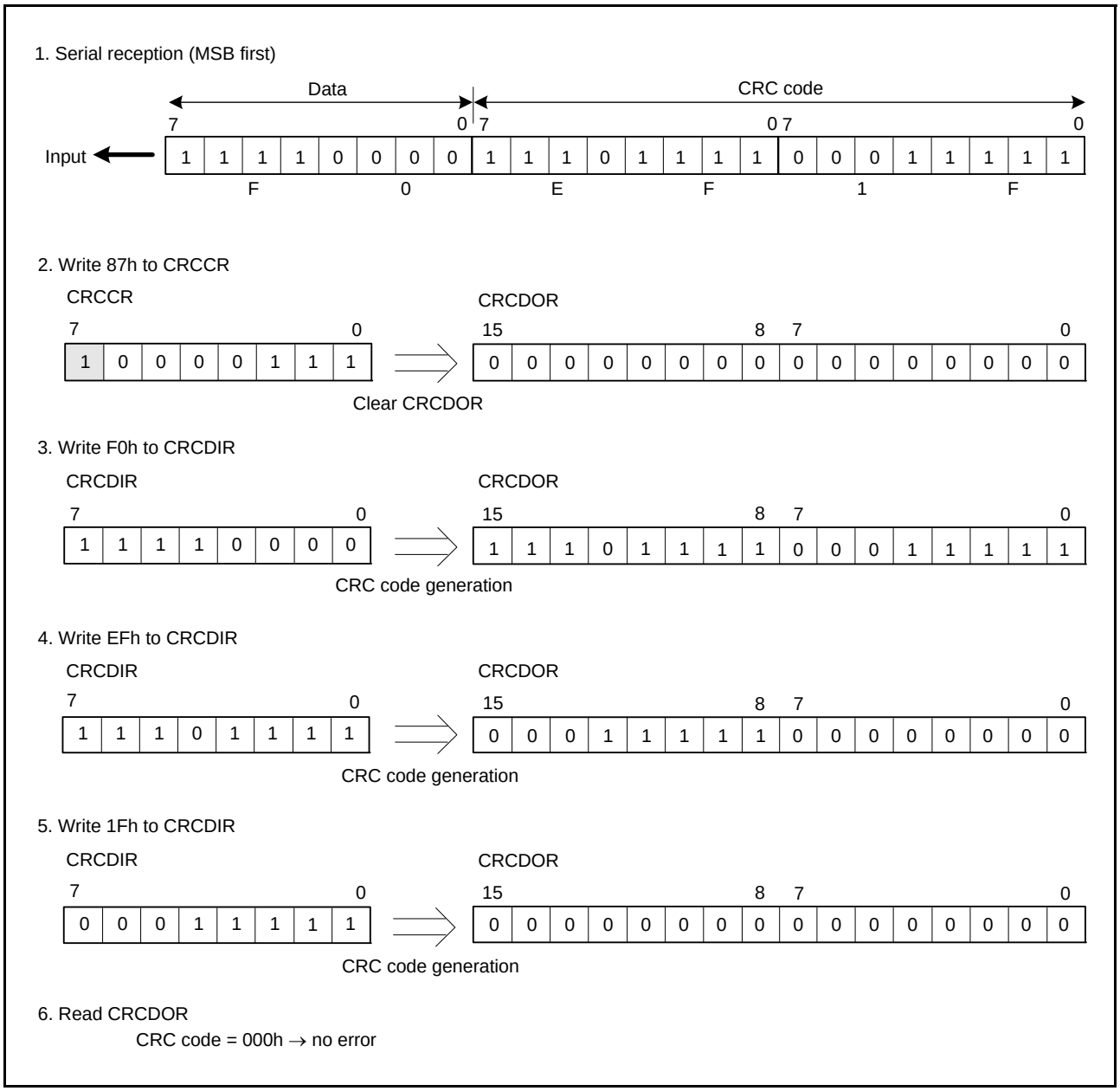


Figure 39.5 MSB-First Data Reception

39.4 Usage Notes

39.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). The initial setting is for operation of the CRC calculator to be stopped. Register access is enabled by clearing the module stop state. For details, see section 11, Low Power Consumption.

39.5 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB-first or MSB-first.

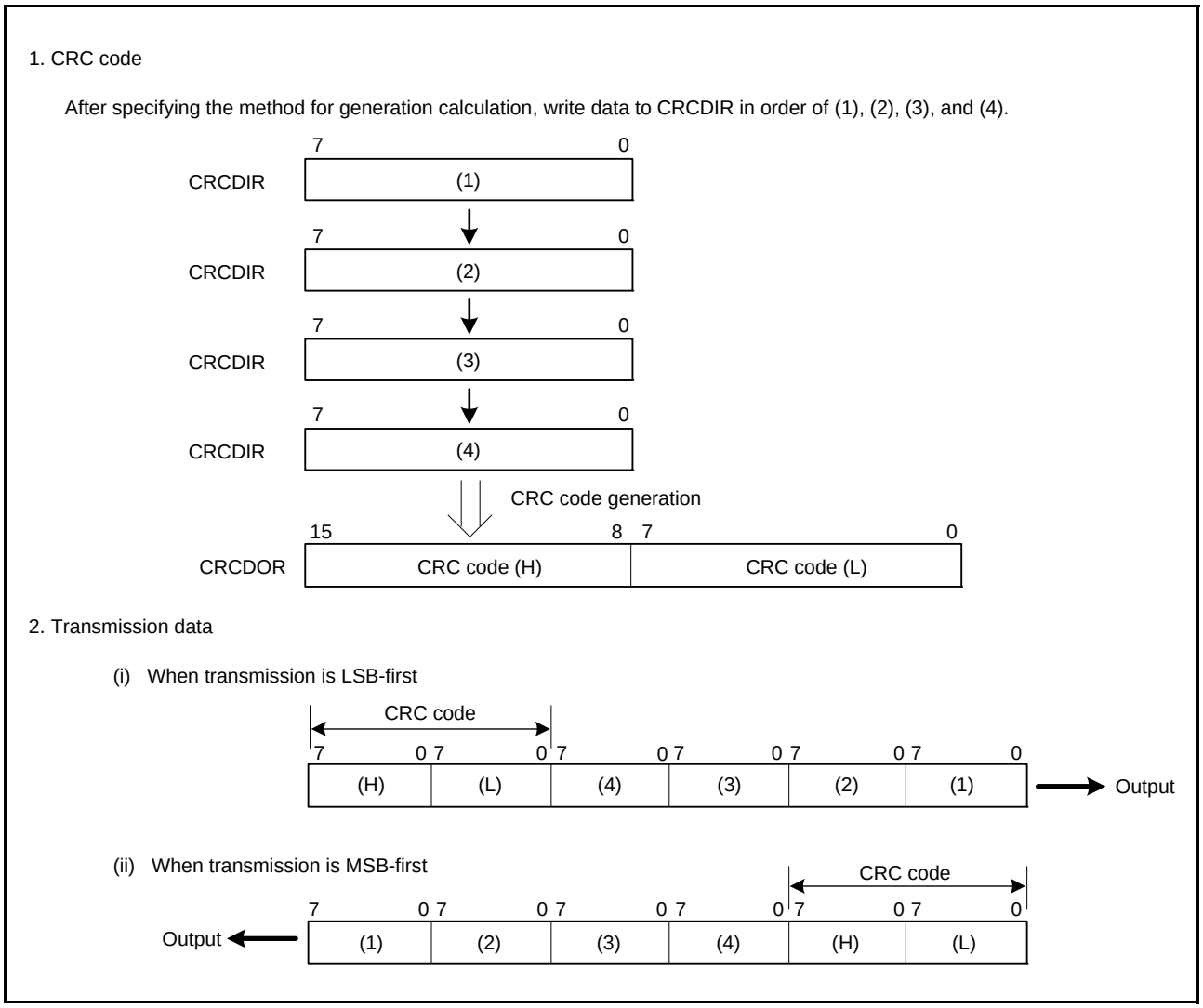


Figure 39.6 LSB-First and MSB-First Data Transmission

40. 12-Bit A/D Converter (S12ADa)

40.1 Overview

The RX63N/RX631 Group includes a 12-bit successive approximation A/D converter. Up to 21 channel analog inputs, temperature sensor outputs, or A/D internal reference voltages can be selected.
The A/D converter has two operating modes: single scan mode in which the analog input of up to 21 channels is converted for only once in ascending channel order and continuous scan mode in which the analog inputs of up to 21 arbitrarily selected channels are continuously converted in ascending channel order.
A/D conversion of the temperature sensor output or the A/D internal reference voltage is accomplished independently.
Table 40.1 lists the specifications of the 12-bit A/D converter and Table 40.2 indicates the functions of the 12-bit A/D converter. Figure 40.1 shows a block diagram of the 12-bit A/D converter.

Table 40.1 Specifications of 12-Bit A/D Converter

Item	Specifications
Number of units	One unit
Input channels	Up to 21 channels
Extended analog inputs	Temperature sensor output, A/D internal reference voltage*3
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	1.0 μs per 1 channel (when operating with peripheral module clock PCLK = 50 MHz)
A/D conversion clock (ADCLK)	4 types: PCLK, PCLK/2, PCLK/4, PCLK/8
Data registers	• For analog input: 21 data registers • For temperature sensor: One data register • For internal reference voltage: One data register • The A/D conversion result is held in a 12-bit A/D data register. • In A/D-converted value addition mode, A/D conversion results are stored in a 14-bit A/D data register.
Operating modes	• Single scan mode: A/D conversion is to be performed for only once on the analog inputs of up to 21 arbitrarily selected channels. A/D conversion is to be performed only once on the temperature sensor output. A/D conversion is to be performed only once on the internal reference voltage. • Continuous scan mode: A/D conversion is to be performed sequentially on the analog inputs of up to 21 arbitrarily selected channels.*1
Conditions of A/D conversion start	• Software trigger • Synchronous trigger Trigger by MTU, TPU, or TMR • Asynchronous trigger A/D conversion can be started by the ADTRG0# pin.
Function	• Sample-and-hold function • Number of sampling states is adjustable. • A/D-converted value addition mode
Interrupt source	• A scan end interrupt (S12ADI0) request can be generated on completion of A/D conversion. • A S12ADI0 interrupt can activate DMAC or DTC.
Low power consumption function	Module stop state can be specified.*2

Note 1. Do not use continuous scan mode when temperature sensor output or A/D internal reference voltage is selected.
Note 2. When the module stop state is canceled, A/D conversion can be started after 10 ms has elapsed.
Note 3. Refer to section 45., Electrical Characteristics for the A/D internal reference voltage.

Table 40.2 Function Overview of the 12-Bit A/D Converter

Item			Function/Internal Trigger Source	
Analog input channel			AN000 to AN020	
A/D conversion start conditions	Software	Software trigger	Enabled	
	Asynchronous trigger	ADTRG0#	Enabled	
	Synchronous trigger (MTU, TMR, TPU)*3	TRG0AN_0	MTU0.TGRA and MTU0.TCNT	Input-capture/compare-match
		TRG0BN_0	MTU0.TGRB and MTU0.TCNT	
		TRGAN_0	MTU0.TGRA and MTU0.TCNT	Input-capture/compare-match
			MTU1.TGRA and MTU1.TCNT	
			MTU2.TGRA and MTU2.TCNT	
			MTU3.TGRA and MTU3.TCNT	
			MTU4.TGRA and MTU4.TCNT	
			MTU4.TCNT	TCNT underflow (trough) in complementary PWM mode
		TRGAN_1	TPU0.TGRA	Input-capture/compare-match
			TPU1.TGRA	
			TPU2.TGRA	
			TPU3.TGRA	
			TPU4.TGRA	
		TRG0EN_0	MTU0.TGRE and MTU0.TCNT	Compare-match
		TRG0FN_0	MTU0.TGRF and MTU0.TCNT	
		TRG4ABN_0	MTU4.TADCORA and MTU4.TCNT or MTU4.TADCORB and MTU4.TCNT	Compare-match using delayed A/D conversion start request function
		TRG4ABN_1	TPU0.TGRA	Input-capture/compare-match
		TMTRG0AN_0	TMR0.TCORA and TMR0.TCNT	Compare-match
		TMTRG0AN_1	TMR2.TCORA and TMR2.TCNT	
Interrupt			S12ADI0 interrupt	
Module stop function setting*1,*2			MSTPCRA.MSTPA17 bit	

Note 1. When the module stop state is canceled, A/D conversion can be started after 10 ms has elapsed.
Note 2. For details, see section 11, Low Power Consumption.
Note 3. The suffix "_0" or "_1" appended to the name of a synchronous trigger signal indicates the number of the unit. For synchronous trigger output settings, see section 22.4.3, A/D Converter Activation, and section 26.6.2, A/D Converter Activation.

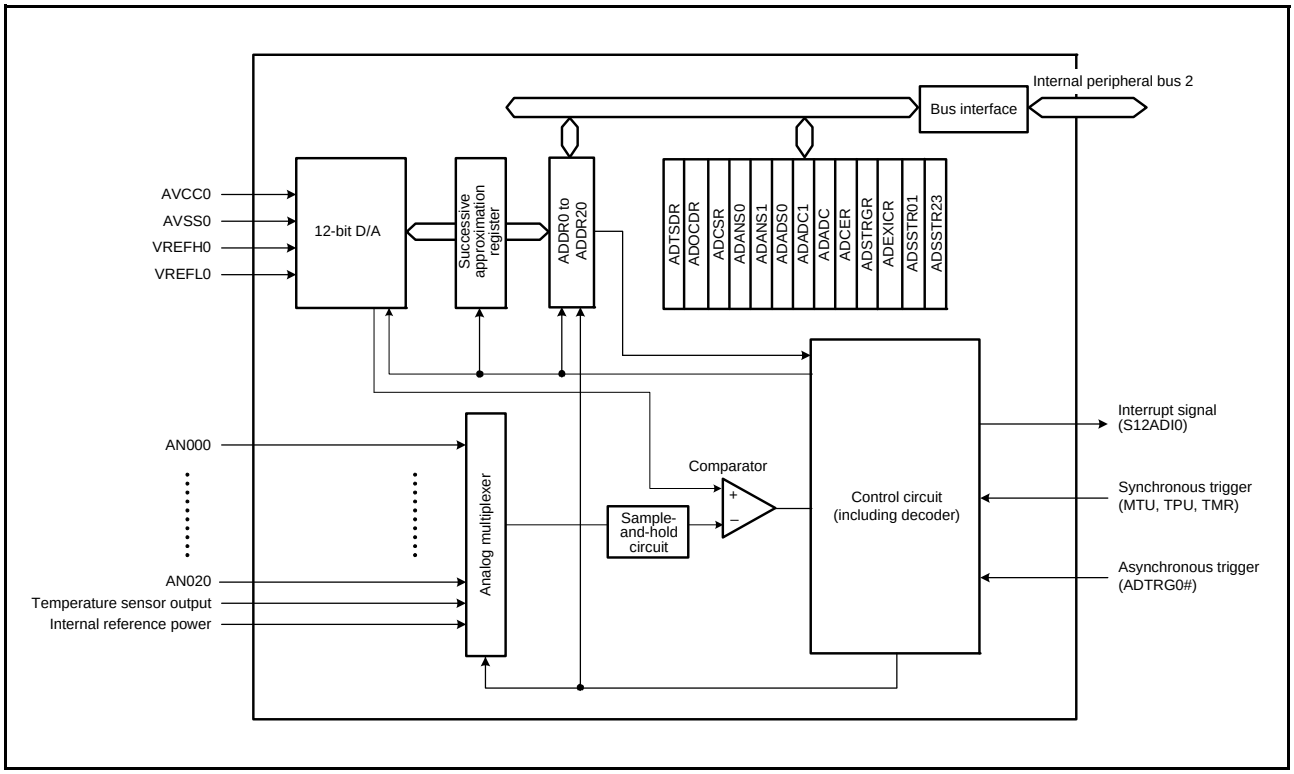


Figure 40.1 Block Diagram of 12-Bit A/D Converter

Table 40.3 indicates the input pins of the 12-bit A/D converter.

Table 40.3 Input Pins of 12-Bit A/D Converter

Pin Name	Input	Function
AVCC0	Input	Analog block power supply pin for 12-bit A/D converter
AVSS0	Input	Ground pin for 12-bit A/D converter
VREFH0	Input	Reference power supply pin for 12-bit A/D converter
VREFL0	Input	Reference power supply ground pin for 12-bit A/D converter
AN000 to AN020	Input	Analog input pins
ADTRG0#	Input	External trigger input pin for starting A/D conversion

40.2 Register Descriptions

40.2.1 A/D Control Register (ADCSR)

Address(es): 0008 9000h

	b7	b6	b5	b4	b3	b2	b1	b0
	ADST	ADCS	—	ADIE	CKS[1:0]		TRGE	EXTRG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXTRG	Trigger Select*1	0: A/D conversion is started by the synchronous trigger (MTU, TPU, or TMR). 1: A/D conversion is started by the asynchronous trigger (ADTRG0#).	R/W
b1	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
b3, b2	CKS[1:0]	A/D Conversion Clock Select	b3 b2 0 0: PCLK/8 0 1: PCLK/4 1 0: PCLK/2 1 1: PCLK	R/W
b4	ADIE	Scan End Interrupt Enable	0: Disables S12ADI0 interrupt generation upon scan completion. 1: Enables S12ADI0 interrupt generation upon scan completion.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ADCS	Scan Mode Select	0: Single scan mode 1: Continuous scan mode	R/W
b7	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)
If 1 is written to both the TRGE and EXTRG bits in ADCSR when a high-level signal is input to the external pin (ADTRG0#), and then if the ADTRG0# signal is driven low, the falling edge of ADTRG0# is 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 PCLK cycles.

CKS[1:0] Bits (A/D Conversion Clock Select)

The CKS[1:0] bits select the clock to be used in A/D conversion.

The CKS[1:0] bits should be set with the ADST bit being 0. (In addition, the CKS[1:0] bits should not be set simultaneously with the setting of the ADST bit to 1.)

ADIE Bit (Scan Conversion End Interrupt Enable)

The ADIE bit enables or disables the scan end interrupt (S12ADI0).

When A/D conversion of all the selected channels is completed while the ADIE bit is set to 1, the scan conversion end interrupt is generated. When A/D conversion is completed even if the temperature sensor output or the A/D internal reference voltage is selected and the ADIE bit is set to 1, the scan end interrupt is generated.

ADCS Bit (Scan Mode Select)

The ADCS bit selects the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 21 channels selected with the ADANS0 and ADANS1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, scan conversion is stopped.

In continuous scan mode, while the ADST bit in ADCSR is 1, A/D conversion is performed for the analog inputs of a maximum of 21 channels selected with the ADANS0 and ADANS1 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 beginning at the first channel. A/D conversion is repeated until the ADST bit in ADCSR is set to 0.

When the temperature sensor output or the A/D internal reference voltage is selected, single scan mode should be selected and all the channels selected by the ADANS0 and ADANS1 registers should be deselected, after which A/D conversion is to be started. The A/D conversion stops after completion of A/D conversion of the temperature sensor output or the A/D internal reference voltage selected.

The ADCS bit should be set while the ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit.)

Table 40.4 shows the scope of selectability of items for A/D conversion by scan mode.

Table 40.4 Scan Mode and Scope of Selectability for A/D Conversion

	Scope of A/D Conversion			
Scan Mode Setting	**Self-Diagnosis**	**Analog Input**	**Temperature Sensor Output**	**Internal Reference Voltage**
Single scan mode	○	○	○	○
Continuous scan mode	○	○	×	×

○: Selectable, ×: Not selectable

Note 1. Ensure that the analog input is not selected at the same time as the temperature sensor output or A/D internal reference voltage.

Note 2. Ensure that neither the temperature sensor output nor the A/D internal reference voltage is selected at the same time as the analog input.

ADST Bit (A/D Conversion Start)

This 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]

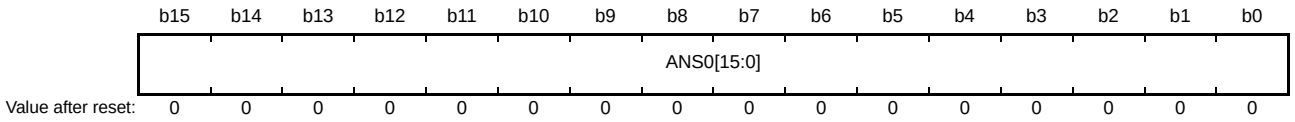
- When 1 is written by software
- When the ADCSR.EXTRG bit is set to 0, the ADCSR.TRGE bit is set to 1, and the synchronous trigger (MTU, TPU, or TMR) selected by the ADSTRGR.ADSTRS[3:0] bits is detected
- When the ADCSR.EXTRG and ADCSR.TRGE bits are set to 1, the ADSTRGR.ADSTRS[3:0] bits are set to “0000b”, and the asynchronous trigger is detected

[Clearing conditions]

- When 0 is written by software
- When the A/D conversion of all channels selected in single scan mode is completed
- When the A/D conversion of the temperature sensor output or the A/D internal reference voltage selected in single scan mode is completed

40.2.2 A/D Channel Select Register 0 (ADANS0)

Address(es): 0008 9004h



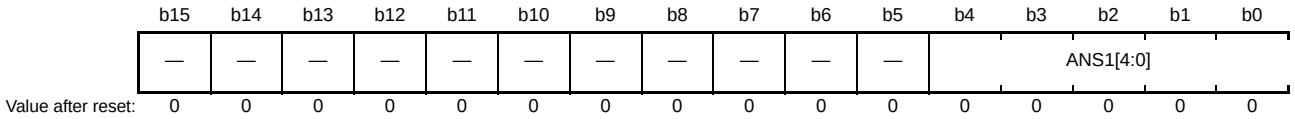
Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ANS0[15:0]	A/D Conversion Channel Select	0: AN000 to AN015 are not subjected to conversion. 1: AN000 to AN015 are subjected to scan conversion.	R/W

ANS0[15:0] Bits (A/D Conversion Channel Select)

These bits select analog inputs of the channels AN000 to AN015 that are subjected to A/D conversion. The channel that is to be selected and the number of channels can be arbitrarily set. ANS0[0] corresponds to AN000 and ANS0[15] corresponds to AN015. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the temperature sensor output and the A/D internal reference voltage should not be performed.
The ANS0[15:0] bits should be set while the ADST bit in ADCSR is set to 0.

40.2.3 A/D Channel Select Register 1 (ADANS1)

Address(es): 0008 9006h



Bit	Symbol	Bit Name	Description	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4 to b0	ANS1[4:0]	A/D Conversion Channel Select	0: AN016 to AN020 are not subjected to conversion. 1: AN016 to AN020 are subjected to scan conversion.	R/W

ANS1[4:0] Bits (A/D Conversion Channel Select)

These bits select analog inputs of the channels AN016 to AN020 that are subjected to A/D conversion. The channel that is to be selected and the number of channels can be arbitrarily set. ANS1[0] corresponds to AN016 and ANS1[4] corresponds to AN020. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the temperature sensor output and the A/D internal reference voltage should not be performed.
The ANS1[4:0] bits should be set while the ADST bit in ADCSR is set to 0.

40.2.4 A/D-Converted Value Addition Mode Select Register 0 (ADADS0)

Address(es): 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADS0[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	ADS0[15:0]	A/D-Converted Value Addition Channel Select	0: A/D-converted value addition mode for AN000 to AN015 is not selected. 1: A/D-converted value addition mode for AN000 to AN015 is selected.	R/W

ADS0[15:0] Bits (A/D-Converted Value Addition Channel Select)

When the ADS0[n] bit of the number that is the same as that of A/D converted channel selected by ANS0[n] (n = 0 to 15) in ADANS is set to 1, these bits perform A/D conversion of analog input of the selected channels successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the added (integrated) conversion results to the A/D data register. For the channel for which the A/D conversion is performed and addition mode is not selected, a normal one-time conversion is performed and the conversion result is returned to the A/D data register.
The ADS0[15:0] bits should be set while the ADST bit in ADCSR is set to 0.

40.2.5 A/D-Converted Value Addition Mode Select Register 1 (ADADS1)

Address(es): 0008 9008h

	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
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4 to b0	ADS1[4:0]	A/D-Converted Value Addition Channel Select	0: A/D-converted value addition mode for AN016 to AN020 is not selected. 1: A/D-converted value addition mode for AN016 to AN020 is selected.	R/W

ADS1[4:0] Bits (A/D-Converted Value Addition Channel Select)

When the ADS1[n] bit of the number that is the same as that of A/D converted channel selected by ANS1[n] (n = 0 to 4) in ADANS is set to 1, these bits perform A/D conversion of analog input of the selected channels successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the added (integrated) conversion results to the A/D data register. For the channel for which the A/D conversion is performed and addition mode is not selected, a normal one-time conversion is performed and the conversion result is returned to the A/D data register.
The ADS1[4:0] bits should be set while the ADST bit in ADCSR is set to 0.

Figure 40.2 shows a scanning operation sequence in which both the ADS0[2] and ADS0[6] bits are set to 1.
In continuous scan mode (ADCSR.ADCS = 1), it is assumed that the addition count is set to 4 (ADADC.ADC[1:0] = 11b) and the channels AN000 to AN007 are selected (ADANS0.ANS0[15:0] = 00FFh). 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 A/D data register 6. After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000.
For the channel for which the addition mode is not selected, the A/D data register format is determined by the ADRFMT bit in ADCER (flush-right or flush-left).

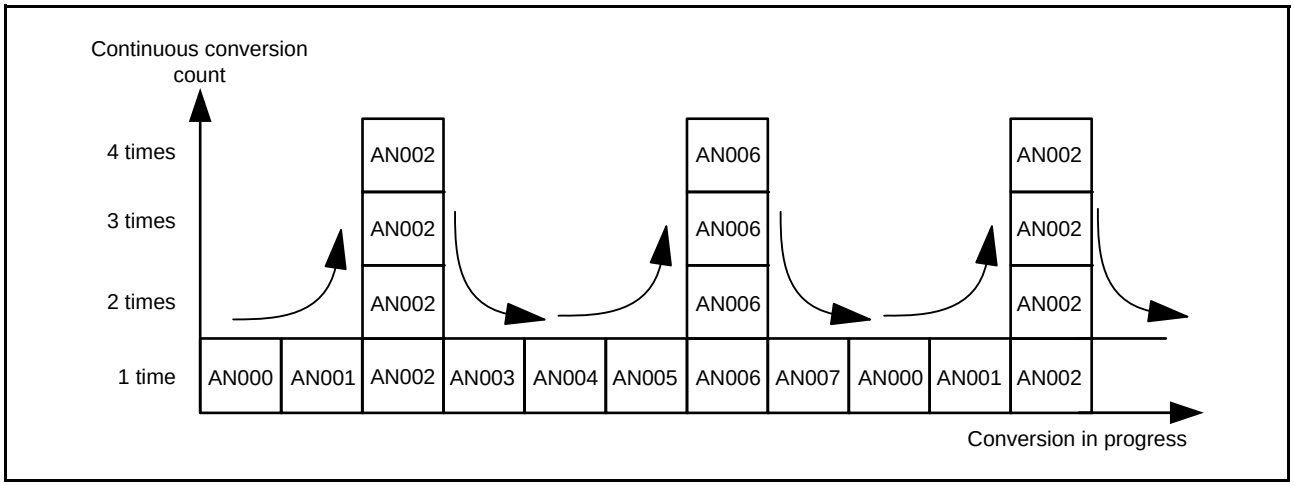


Figure 40.2 Scan Conversion Sequence with ADADC.ADC[1:0] = 11b, ADS0[2] = 1, and ADS0[6] = 1

40.2.6 A/D-Converted Value Addition Count Select Register (ADADC)

Address(es): 0008 900Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	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: 4-time conversion (addition three times)	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADC[1:0] Bits (Addition Count Select)

The ADC[1:0] bits set the addition count common for the channels for which A/D conversion is performed and the addition mode is selected, and for A/D conversion of the temperature sensor output and the A/D internal reference voltage. The ADC[1:0] bits should be set while the ADST bit in ADCSR is set to 0.

40.2.7 A/D Control Extended Register (ADCER)

Address(es): 0008 900Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	—	—	—	—	—	—	ACE	—	—	—	—	—
Value after 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	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	Automatic Clearing Enable	0: Disables automatic clearing. 1: Enables automatic clearing	R/W
b14 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Flush-right is selected for the A/D data register format. 1: Flush-left is selected for the A/D data register format.	R/W

ACE Bit (Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of ADDRy or after the given register has been read by the CPU, DTC, or DMAC. This function also enables detection of failures to update ADDRy.
However, neither the A/D temperature sensor data register (ADTSDR) nor the A/D internal reference voltage data register (ADOCDR) is automatically cleared.

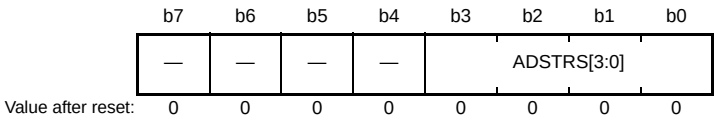
ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDRy, the A/D temperature sensor data register (ADTSDR), and the A/D internal reference voltage data register (ADOCDR).
When the A/D converted value addition mode is selected, the format of each data register is fixed to flush-left, irrespective of the ADRFMT bit value.

For details on the format of the data registers, see section 40.2.12, A/D Data Register y (ADDRy) (y = 0 to 20), section 40.2.10, A/D Temperature Sensor Data Register (ADTSDR), and section 40.2.11, A/D Internal Reference Voltage Data Register (ADOCDR).

40.2.8 A/D Start Trigger Select Register (ADSTRGR)

Address(es): 0008 9010h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ADSTRS[3:0]	A/D Conversion Start Trigger Select	These bits select the A/D conversion startup source from the on-chip peripheral modules.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSTRS[3:0] Bits (A/D Conversion Start Trigger Select)

The ADSTRS[3:0] bits select A/D conversion start trigger. Table 40.5 lists a list of A/D conversion startup sources.

- When using the A/D conversion startup source of the synchronous trigger (MTU, TPU, or TMR), set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 0.
- When using the asynchronous trigger (ADTRG0#), set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 1.
- Software trigger (ADST bit in ADCSR) is always enabled regardless of the set values of the TRGE, EXTRG in ADCSR, and the ADSTRS[3:0] bits.

Table 40.5 List of Sources to Start A/D Conversion

Selectable Source	ADSTRS[3]	ADSTRS[2]	ADSTRS[1]	ADSTRS[0]	Remarks
Software trigger	—	—	—	—	
Asynchronous trigger	0	0	0	0	
TRG0AN_0	0	0	0	1	MTU0.TGRA and MTU0.TCNT
TRG0BN_0	0	0	1	0	MTU0.TGRB and MTU0.TCNT
TRGAN_0	0	0	1	1	MTUn.TGRA and MTUn.TCNT (n = 0 to 4)
TRGAN_1	0	1	0	0	TPUn.TGRA (n = 0 to 4)
TRG0EN_0	0	1	0	1	MTU0.TGRE and MTU0.TCNT
TRG0FN_0	0	1	1	0	MTU0.TGRF and MTU0.TCNT
TRG04ABN_0	0	1	1	1	MTU4.TADCORA and MTU4.TCNT or MTU4.TADCORB and MTU4.TCNT
TRG04ABN_1	1	0	0	0	TPU0.TGRA
TMTRG0AN_0	1	0	0	1	TMR0.TCORA and TMR0.TCNT
TMTRG0AN_1	1	0	1	0	TMR2.TCORA and TMR2.TCNT

40.2.9 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): 0008 9012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OCS	TSS	—	—	—	—	—	—	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 Mode Select	0: Temperature sensor output A/D converted value addition mode is not selected 1: Temperature sensor output A/D converted value addition mode is selected	R/W
b2	OCSAD	A/D Internal Reference Voltage A/D Converted Value Addition Mode Select	0: A/D Internal reference voltage A/D converted value addition mode is not selected 1: A/D Internal reference voltage A/D converted value addition mode is selected	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSS	Temperature Sensor Output A/D Conversion Select	0: A/D conversion of temperature sensor output is not performed 1: A/D conversion of temperature sensor output is performed	R/W
b9	OCS	A/D Internal Reference Voltage A/D Conversion Select	0: A/D conversion of A/D internal reference voltage is not performed 1: A/D conversion of A/D internal reference voltage is performed	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TSSAD Bit (Temperature Sensor Output A/D Converted Value Addition Mode Select)

The TSSAD bit selects A/D conversion for the temperature sensor output. Setting the TSSAD bit to 1 performs A/D conversion of the temperature sensor output successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the integrated value to the A/D temperature sensor data register (ADTSDR).
The TSSAD bit should be set while the ADST bit in ADCSR is set to 0.

OCSAD Bit (A/D Internal Reference Voltage A/D Converted Value Addition Mode Select)

The OCSAD bit selects A/D conversion for the A/D internal reference voltage. Setting the OCSAD bit to 1 performs A/D conversion of the A/D internal reference voltage successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the integrated value to the A/D internal reference voltage data register (ADOCDR). The OCSAD bit should be set while the ADST bit in ADCSR is set to 0.

TSS Bit (Temperature Sensor Output A/D Conversion Select)

The TSS bit selects A/D conversion for the temperature sensor output. When A/D conversion of the temperature sensor output is to be performed, all the bits in the ADANS0 and ADANS1 registers, and the OCS bit should be set to 0. The continuous scan mode (the ADCS bit in ADCSR = 1) should not be selected. The TSS bit should be set while the ADST bit in ADCSR is set to 0.

OCS Bit (A/D Internal Reference Voltage A/D Conversion Select)

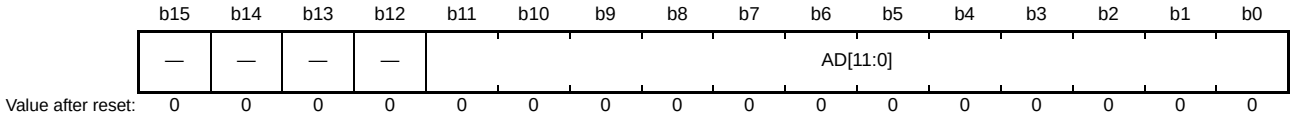
The OCS bit selects A/D conversion for the A/D internal reference voltage. When A/D conversion of the internal reference voltage is to be performed, all the bits in the ADANS0 and ADANS1 registers, and the TSS bit should be set to 0. The continuous scan mode (the ADCS bit in ADCSR = 1) should not be selected. The OCS bit should be set while the ADST bit in ADCSR is set to 0.

40.2.10 A/D Temperature Sensor Data Register (ADTSDR)

ADTSDR is a 16-bit read-only register that stores the A/D conversion results of the temperature sensor output.
The following different formats are used depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for flush-right)

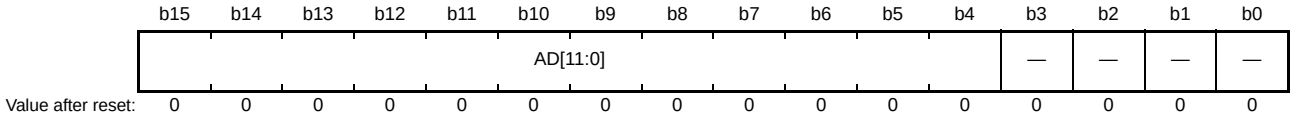
Address(es): 0008 901Ah



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	—	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

- ADCER.ADRFMT = 1 (Setting for flush-left)

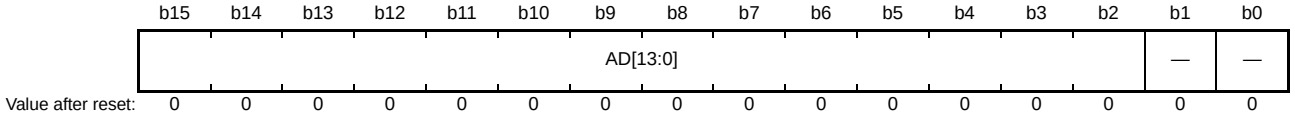
Address(es): 0008 901Ah



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]	—	12-bit A/D-converted value	R

- When A/D-converted value addition mode is selected

Address(es): 0008 901Ah



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]	—	14-bit A/D-converted value addition result	R

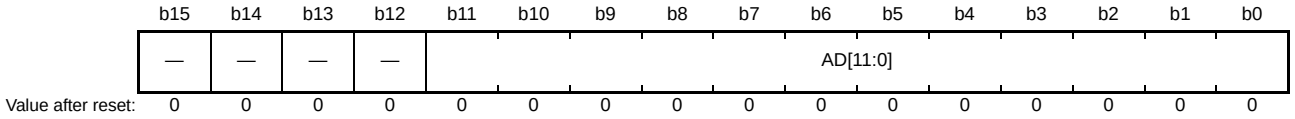
When A/D-converted value addition mode is selected, the AD[13:0] bits in ADTSDR show the value added by the A/D-converted value of the temperature sensor output. In A/D-converted value addition mode, the setting of ADRFMT bit in ADCER becomes invalid and the format of the register becomes flush-left.

40.2.11 A/D Internal Reference Voltage Data Register (ADOCDR)

ADOCDR is a 16-bit read-only register that stores the A/D conversion results of the A/D internal reference voltage. The following different formats are used depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for flush-right)

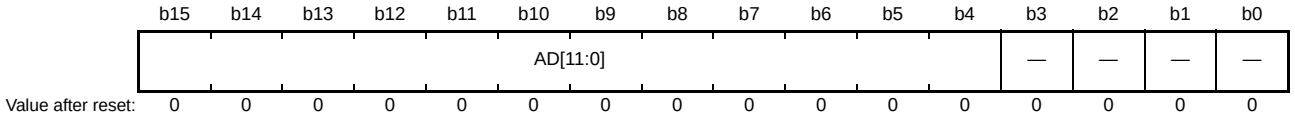
Address(es): 0008 901Ch



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	—	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

- ADCER.ADRFMT = 1 (Setting for flush-left)

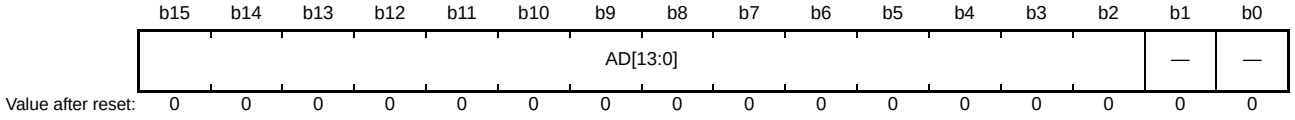
Address(es): 0008 901Ch



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]	—	12-bit A/D-converted value	R

- When A/D-converted value addition mode is selected

Address(es): 0008 901Ch



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]	—	14-bit A/D-converted value addition result	R

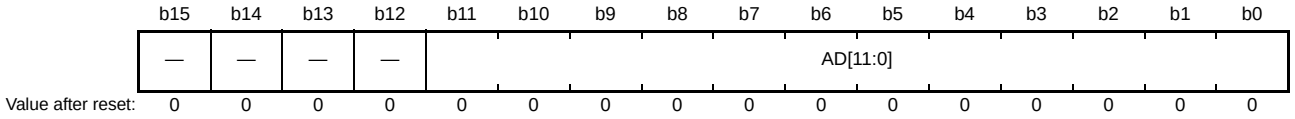
When A/D-converted value addition mode is selected, the AD[13:0] bits in ADOCDR show the value added by the A/D-converted value of the A/D internal reference voltage. In A/D-converted value addition mode, the setting of the ADRFMT bit in ADCER becomes invalid and the format of the register becomes flush-left.

40.2.12 A/D Data Register y (ADDRy) (y = 0 to 20)

A/D data registers 0 to 20 (ADDR0 to ADDR20) are 16-bit read-only registers which store the A/D conversion results of channels AN000 to AN020.
The A/D data registers use the following different formats depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for flush-right)

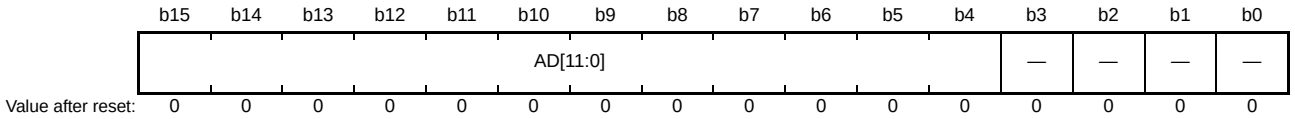
Address(es): 0008 9020h to 0008 9048h



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	—	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

- ADCER.ADRFMT = 1 (Setting for flush-left)

Address(es): 0008 9020h to 0008 9048h



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]	—	12-bit A/D-converted value	R

- When A/D-converted value addition mode is selected

Address(es): 0008 9020h to 0008 9048h

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]	—	14-bit A/D-converted value addition result	R

When A/D-converted value addition mode is selected, the AD[13:0] bits in ADDRy show the value added by the A/D-converted value of the respective channels. In A/D-converted value addition mode, the setting of the ADRFMT bit in ADCER becomes invalid and the format of the register becomes flush-left.

The following minimum and maximum values apply to channels on which A/D-converted value addition mode is selected.

First conversion: 0000h ≤ ADDRy (y = 0 to 20) ≤ 3FFCh
(ADDRy (y = 0 to 20): Bits 15 and 14 = 00b, bits 13 to 2 = AD11 to AD0, bits 1 and 0 = 00b)
Second conversion: 0000h ≤ ADDRy (y = 0 to 20) ≤ 7FF8h
(ADDRy (y = 0 to 20): Bit 15 = 0b, bits 14 to 2 = AD12 to AD0, bits 1 and 0 = 00b)
Third conversion: 0000h ≤ ADDRy (y = 0 to 20) ≤ BFF4h
(ADDRy (y = 0 to 20): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)
Fourth conversion: 0000h ≤ ADDRy (y = 0 to 20) ≤ FFF0h
(ADDRy (y = 0 to 20): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

40.2.13 A/D Sampling State Register 01 (ADSSTR01)

Address(es): 0008 9060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SST1[7:0]							
Value after reset:	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SST1[7:0]	Sampling Time 1 Setting	These bits set the sampling time as a number of states (cycles of the A/D conversion clock) from 10 to 255.	R/W
b15 to b8	—	Reserved	The write value should be 00010100b. These bits are read as the same value.	R/W

SST1[7:0] Bits (Sampling Time 1 Setting)

These bits are used to set the sampling time for the analog input of the selected channel. One state refers to one cycle of the A/D conversion clock (ADCLK), so when the frequency of the clock is 50 MHz, one state becomes 20 ns. The default setting is 20 states. The sampling time can be adjusted if the source impedance of the analog input signal is high enough that the sampling time is too short or the ADCLK frequency is too low for the current setting. Only set the SST1[7:0] bits while the ADSCR.ADST bit is 0.
The sampling time setting should be in the range from 10 to 255 states, so that the sampling time is longer than 0.4 µs.

40.2.14 A/D Sampling State Register 23 (ADSSTR23)

Address(es): 0008 9070h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SST2[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These write value should be 00010100b. These bits are read as the same value.	R/W
b15 to b8	SST2[7:0]	Sampling Time 2 Setting	These bits set the sampling time in 10 to 255 states.	R/W

SST2[7:0] Bits (Sampling Time 2 Setting)

These bits set the sampling time for the temperature sensor output. One state is one cycle of the A/D conversion clock (ADCLK). When the ADCLK is running at 50 MHz, one state is 20 ns. The initial value is 20 states. If the impedance of the analog input signal source is too high for this to secure sufficient sampling time or if the ADCLK is at a low frequency, the sampling time can be adjusted. The SST2[7:0] bits should be set while the ADCSR.ADST bit is 0.
The sampling time setting for the temperature sensor output should be in the range from 10 to 255 states, so that the sampling time is longer than 0.4 µs.

40.3 Operation

40.3.1 Scanning Operation

In Scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.
A scan conversion is performed in two operating modes: single scan mode and continuous scan mode. In single scan mode, one or more specified channels are scanned once. In the continuous scan mode, one or more specified channels are scanned repeatedly until the ADST bit in ADCSR is set to 0 by software.
In either mode, A/D conversion is performed for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.
A/D conversion is performed in single scan mode, when the temperature sensor output or A/D internal reference voltage is selected. This operation is similar to the scanning operation in which only one channel is selected in single scan mode.

40.3.2 Single Scan Mode

40.3.2.1 Basic Operation

In basic operation of single scan mode, A/D conversion is to be performed for only once on the analog input of the specified single channel as below. In selected channel scanning, temperature sensor output A/D conversion select bit (TSS) and A/D internal reference voltage A/D conversion select bit (OCS) in ADEXICR should both be set to 0 (non-selection).

(1) A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or TMR), or asynchronous trigger input. A/D conversion is performed for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.
(2) When A/D conversion for single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy) of the channel.
(3) When A/D conversion of all selected channels is completed, if the ADCSR.ADIE bit is set to 1 (enables S12ADI0 interrupt generation upon scan conversion completion), an S12ADI0 interrupt request is generated.
(4) The ADCSR.ADST bit remains at 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all selected channels ends. Then the 12-bit A/D converter enters a wait state.

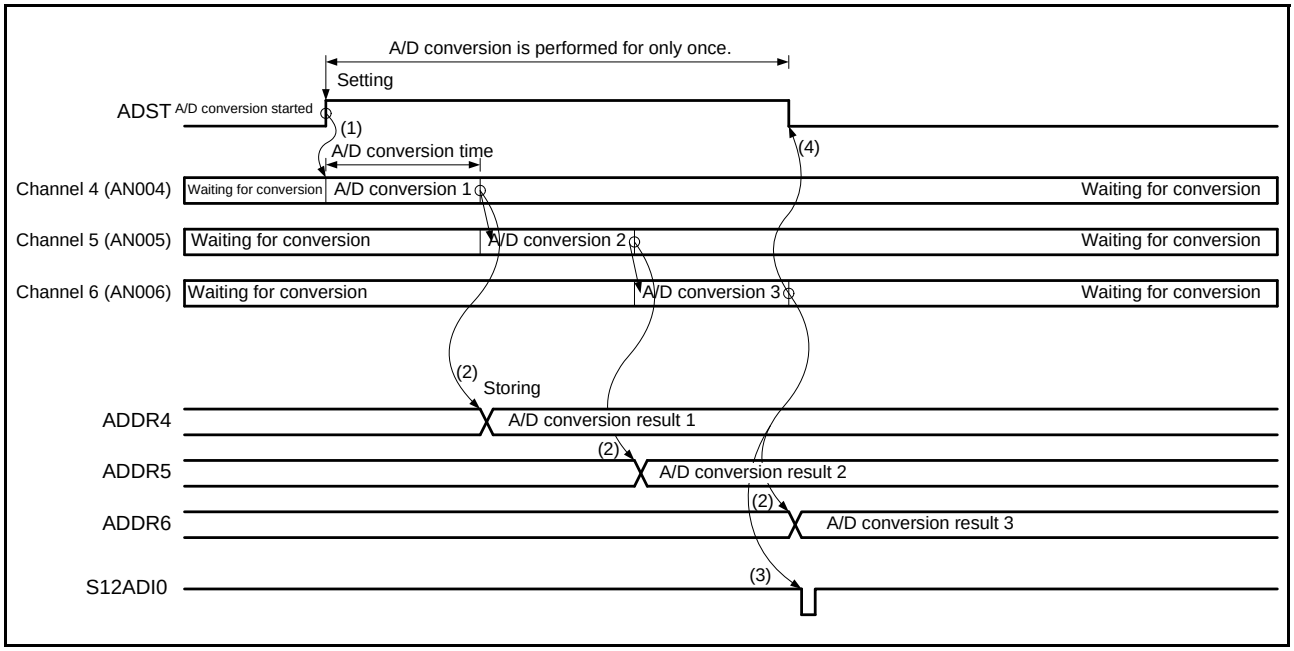


Figure 40.3 Example of Operation of Single Scan Mode (Basic Operation: AN004 to AN006 Selection)

40.3.2.2 A/D Conversion when selecting Temperature Sensor Output/Internal Reference Voltage

When A/D conversion for the temperature sensor output or A/D internal reference voltage is selected, it is performed in single scan mode. The operation is as follows.

Non-selection must be set for all channels (set all bits in ADANS0.ANS0[15:0] and ADANS1.ANS1[4:0] to 0).

Furthermore, when A/D conversion for the temperature sensor output is selected, set A/D internal reference voltage A/D conversion select bit (OCS) in ADEXICR to 0 (non-selection). When A/D conversion of A/D internal reference voltage is selected, set temperature sensor output A/D conversion select bit (TSS) in ADEXICR to 0 (non-selection).

(1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, the synchronous trigger (MTU, TPU, TMR), or the asynchronous trigger input, A/D conversion is started for the temperature sensor output or A/D internal reference voltage.

(2) When A/D conversion is completed, the A/D conversion results are stored into the corresponding A/D temperature sensor data register (ADTSDR) or the A/D internal reference voltage data register (ADOCDR). If the ADIE bit in ADCSR is set to 1 (enables S12ADI0 interrupt generation upon scan conversion completion), an S12ADI0 interrupt request is generated.

(3) The ADST bit in ADCSR remains at 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion ends. Then the 12-bit A/D converter enters a wait state.

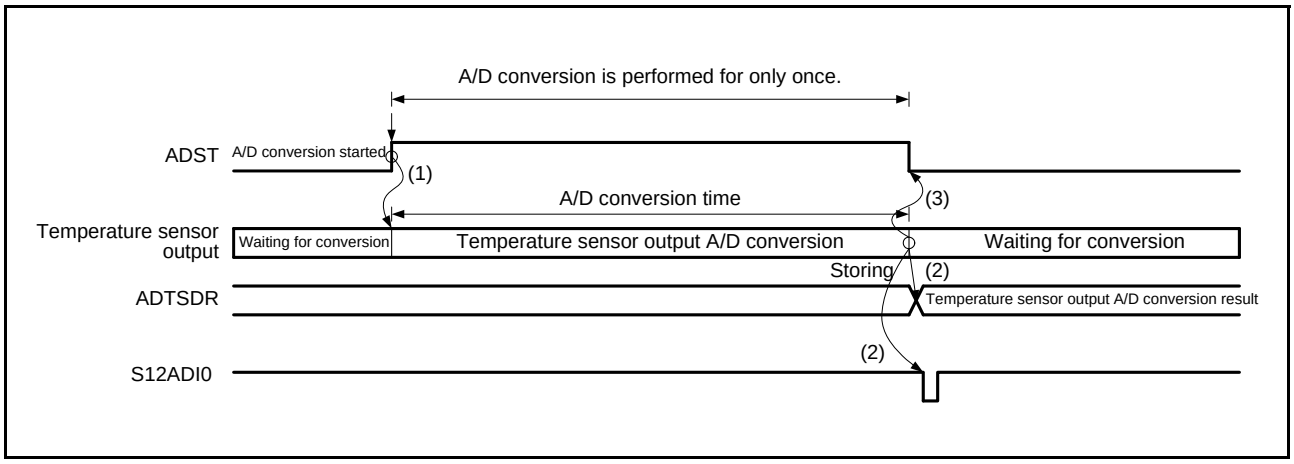


Figure 40.4 Example of Single Scan Mode (Temperature Sensor Output Selection)

40.3.3 Continuous Scan Mode

40.3.3.1 Basic Operation

In basic operation of continuous scan mode, A/D conversion is to be performed sequentially on the analog inputs of the specified channels as below.

In continuous scan mode, temperature sensor output A/D conversion select bit (TSS) and A/D internal reference voltage A/D conversion select bit (OCS) in ADEXICR should both be set to 0 (non-selection).

(1) A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or TMR), or asynchronous trigger input. A/D conversion is performed for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.

(2) When A/D conversion for the 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 selected channels is completed, if the ADIE bit in ADCSR is set to 1 (enables S12ADI0 interrupt generation upon scan conversion completion), an S12ADI0 interrupt request is generated. Furthermore, the 12-bit A/D converter continuously starts A/D conversion for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.

(4) The ADST bit in ADCSR is not cleared automatically, and steps 2 to 3 are repeated as long as the ADST bit remains set to 1 (A/D conversion start). When the ADST in ADCSR bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.

(5) If the ADST bit in ADCSR is later set to 1 (A/D conversion start), A/D conversion starts again for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.

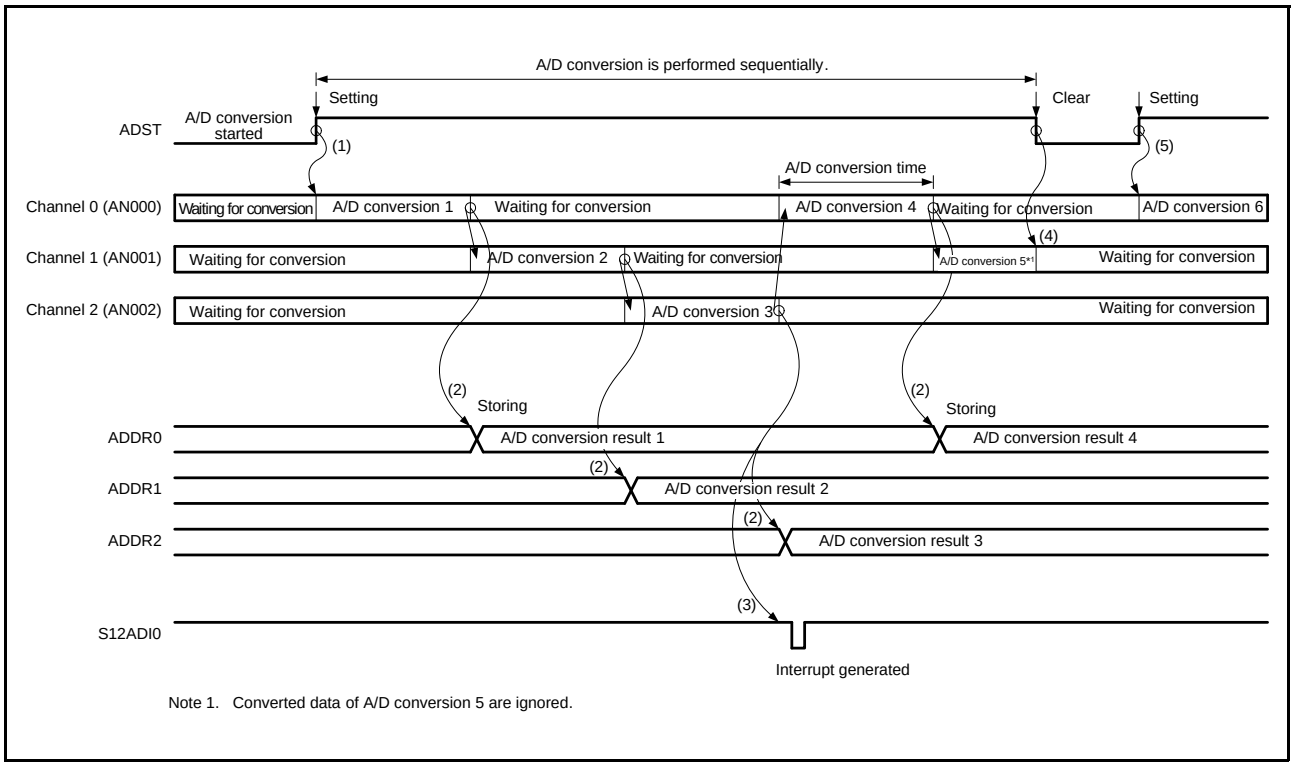


Figure 40.5 Example of Operation of Continuous Scan Mode (Basic Operation: AN000 to AN002 Selection)

40.3.4 Analog Input Scan Time

The times required for the processing listed in Table 40.6 (Table 40.7) determine the scan time.

Table 40.6 Processing During a Scan Operation

Processing	Description
Scan start processing	Processing to start scanning in response to a software trigger, synchronous trigger, or asynchronous trigger
Analog input A/D conversion processing*1	Sample-and-hold and successive-approximation processing by the 12-bit A/D converter
Scan end processing	Processing to store A/D-converted data and output the S12ADI0 interrupt

Note 1. Time for this processing varies with the settings for numbers of sampling states in the SST1[7:0] to SST2[7:0] bit fields in A/D sampling state registers 01 and 23 (ADSSTR01 and ADSSTR23).

Table 40.7 Times Required in Scan Operations (as Numbers of ADCLK and PCLK Cycles)

Item	Symbol	Type or Condition	Cycles
Scan start processing time	t_D	Synchronous trigger (MTU, TPU, TMR) or software trigger	2 PCLK + 3 ADCLK*1
		Asynchronous trigger (ADTRG0#)	4 PCLK + 3 ADCLK*1
A/D conversion processing time	t_{CONV}	ADSSTRxx.SSTy[7:0]*2 Initial value 14h	50 ADCLK
Scan end processing time	t_{ED}	*3	1 PCLK + 2 ADCLK

Note 1. Two and four cycles of the PCLK are the times taken to write to the register and for trigger processing. Three cycles of the ADCLK is the maximum time from setting of the ADCSR.ADST bit to the start of 12-bit A/D converter operations.
Note 2. ADSSTRxx.SSTy[7:0] indicates the ADSSTR01.SST1[7:0], and ADSSTR23.SST2[7:0] bit fields.
Note 3. Two cycles of the ADCLK is the maximum time, and an S12ADI0 interrupt is output one cycle of the PCLK later. For processing time when scanning is forcibly stopped, refer to section 40.5.3, Notes on Restarting A/D Conversion after a Forcible Stop.

The scan conversion time (t_{SCAN}) in single-cycle scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + (t_{CONV} \times n) + t_{ED}$$

- The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single-cycle scan minus t_{ED}.
- The scan conversion time for the second and subsequent cycles in continuous scan mode is a fixed period equal to ($t_{CONV} \times n$).
- A/D conversion processing time (tCONV) in states is given below.
  A/D conversion processing time (tCONV) = 30 (fixed) + setting of the ADSSTRxx.SSTy[7:0] bits

40.3.5 Usage Example of ADDRy Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 selects automatic clearing of each A/D data register (ADDRy) to 0000h when the given register is read by the CPU, DTC, or DMAC.
There is no function for automatically clearing register ADTSDR or ADOCDR.
When the analog input voltage is not in the region of 0 V, the automatic clearing function can be used to detect failure to update the A/D data register y (ADDRy). Since 0000h can be a normal result of A/D conversion when the analog input voltage is in the region of 0 V, the automatic clearing function is not usable.
Examples in which the function to automatically clear the ADDRy registers are enabled and disabled are shown below.
In a case where the ACE bit in ADCER is 0 (automatic clearing is disabled), if the A/D conversion result (0222h) is not written to the ADDRy registers for some reason, the old data (0111h) will become the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan conversion end interrupt, the old data (0111h) can be saved in the general register. When checking data that has not been updated, 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 is 1 (automatic clearing is enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMAC, ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result of 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 conversion 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 whether the read data value is 0000h.

40.3.6 A/D-Converted Value Addition Function

The same channel is A/D converted continuously two to four times and the sum 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.
A/D converted value addition function can be used at the time of selecting the channel selection analog input A/D conversion, temperature sensor output A/D conversion, or A/D internal reference voltage A/D conversion.

40.3.7 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of asynchronous trigger. To start up the A/D conversion by an asynchronous trigger after setting the A/D start trigger select bits (ADSTRS[3:0]) in ADSTRGR to 00h and applying a high-level signal to the asynchronous trigger (ADTRG0# pin), both the TRGE and EXTRG bits in ADCSR should be set to 1. Then if the asynchronous trigger (ADTRG0# pin) is driven low, the ADST bit is set to 1 and the A/D conversion is started. In this case, the low-level pulse width must be at least 1.5 PCLK clock cycles. For the time required for the A/D conversion to start after the ADCSR.ADST bit is set, refer to section 40.5.3, Notes on Restarting A/D Conversion after a Forcible Stop.
Figure 40.6 shows a timing for the ADCSR.ADST bit setting by the asynchronous trigger input.

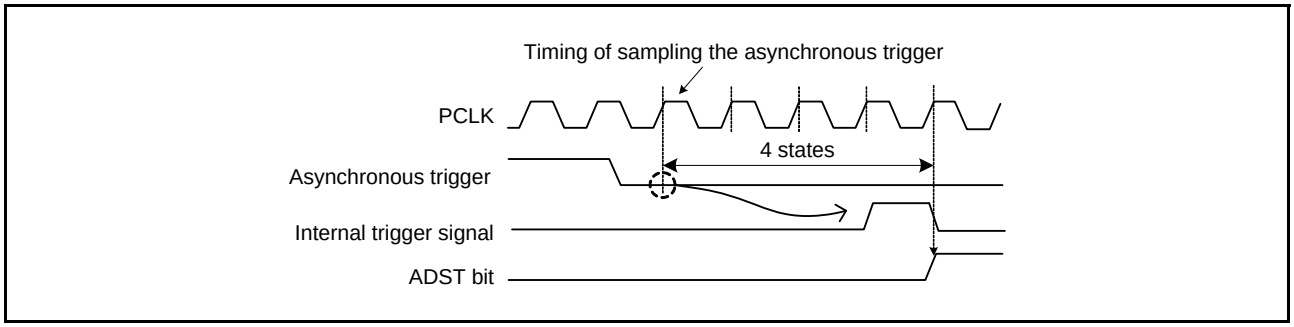


Figure 40.6 Asynchronous Trigger Input Timing

40.3.8 Starting A/D Conversion with Synchronous Trigger from Peripheral Modules

A/D conversion can be started by synchronous trigger of the MTU, TPU, or TMR. To start up the A/D conversion by a synchronous trigger, the TRGE bit in ADCSR should be set to 1, the EXTRG bit in ADCSR should be cleared to 0, and the A/D conversion start source should be selected by ADSTRS[3:0] bits in ADSTRGR.
Table 40.8 lists A/D conversion start sources to be selected.

Table 40.8 A/D Conversion Start Sources to Be Selected

Module	Source	Remarks	ADSTRS[3]	ADSTRS[2]	ADSTRS[1]	ADSTRS[0]
S12AD	ADST	Software trigger	—	—	—	—
External input	ADTRG0#	Asynchronous trigger	0	0	0	0
MTU	TRG1N	TRG0AN_0	0	0	0	1
	TRG2N	TRG0BN_0	0	0	1	0
	TRG3N	TRGAN_0	0	0	1	1
TPU	TRG4N	TRGAN_1	0	1	0	0
MTU	TRG5N	TRG0EN_0	0	1	0	1
	TRG6N	TRG0FN_0	0	1	1	0
	TRG7N	TRG4ABN_0	0	1	1	1
TPU	TRG8N	TRG4ABN_1	1	0	0	0
TMR	TRG9N	TMTRG0AN_0	1	0	0	1
	TRG10N	TMTRG0AN_1	1	0	1	0

Note: • If the ADTRG0# signal is to be set as the trigger for the start of A/D conversion, set the corresponding P0.PDR.B7, P1.PDR.B6, or P2.PDR.B5 bit to 0 (input port operation) and the corresponding P0.PMR.B7, P1.PMR.B6, or P2.PMR.B5 pin to 1 (enabling the input buffer for the given pin). For details, see section 20, I/O Ports and section 21, Multi-Function Pin Controller (MPC).

40.3.8.1 Starting A/D Conversion Using MTU TRG0AN_0 and TRG0BN_0

Upon the TGRA input capture/compare match of the MTU0, the trigger signal TRG0AN_0 can be generated and A/D conversion can be started with TRG0AN_0.
Similarly, upon the TGRB input capture/compare match of the MTU0, the trigger signal TRG0BN_0 can be generated and A/D conversion can be started.
Figure 40.7 shows the connection of MTU TRG0AN_0 output and TRG0BN_0 output to the A/D converter.
To start A/D conversion upon the TRGA input capture/compare match of the MTU0, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, and set ADSTRGR.ADSTRS[3:0] to 0001b (source: TRG1N; corresponding trigger: TRG0AN_0).
To start A/D conversion upon the TRGB input capture/compare match of the MTU0, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, and set ADSTRGR.ADSTRS[3:0] to 0010b (source: TRG2N; corresponding trigger: TRG0BN_0).

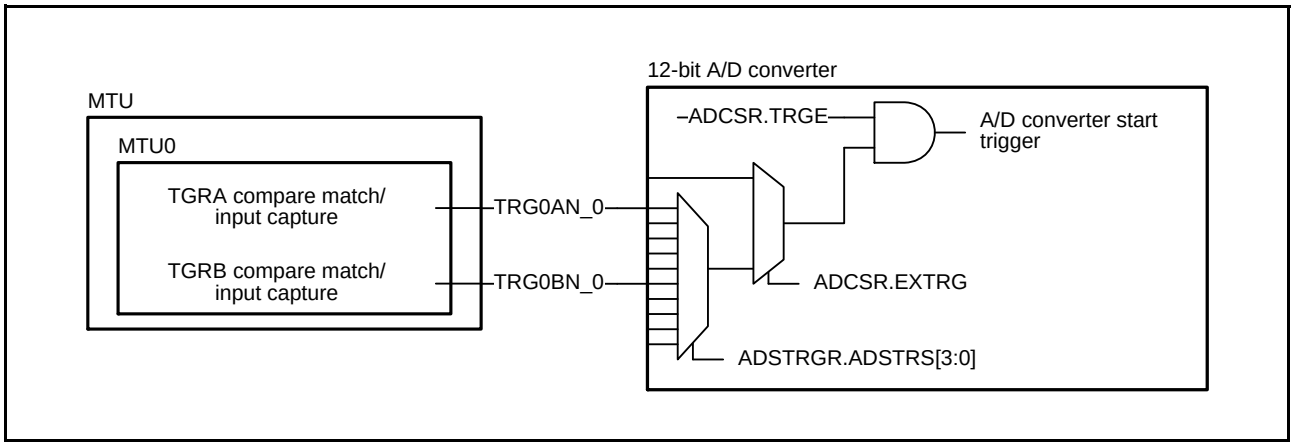


Figure 40.7 Connection of MTU TRG0AN_0 Output and TRG0BN_0 Output to A/D Converter

40.3.8.2 Starting A/D Conversion Using MTU TRGAN_0 and TPU TRGAN_1

Upon the TRGA input capture/compare match of the MTU0 to MTU4 and the TCNT underflow (trough) of the MTU4 in complementary PWM mode, the trigger signal TRGAN_0 can be generated and A/D conversion can be started.
Similarly, upon the TRGA input capture/compare match of the TPU0 to TPU4 in the TPU (unit 0), the trigger signal TRGAN_1 can be generated and A/D conversion can be started.
Figure 40.8 shows the connection of MTU TRGAN_0 output and TPU (unit 0) TRGAN_1 output to the A/D converter.
When A/D conversion is to be started by input capture to or compare match with TGRA in MTU0 to MTU4 and underflows (troughs) of TCNT of MTU4 in complementary PWM mode, set the ADCSR.TRGE bit to 1, the ADCSR.EXTRG bit to 0, the ADSTRGR.ADSTRS[3:0] bits to 0011b (selecting TRG3N as the source and TRGAN_0 as the corresponding trigger), and the MTUn.TIER.TTGE bits (n = 0 to 4) and MTU4.TIER.TTGE2 bit to 1.
Also, when A/D conversion is to be started by input capture to or compare match with TGRA in TPU0 to TPU4, set the ADCSR.TRGE bit to 1, the ADCSR.EXTRG bit to 0, the ADSTRGR.ADSTRS[3:0] bits to 0100b (selecting TRG4N as the source and TRGAN_1 as the corresponding trigger), and the TPUnTIER.TTGE bits (n = 0 to 4) to 1.

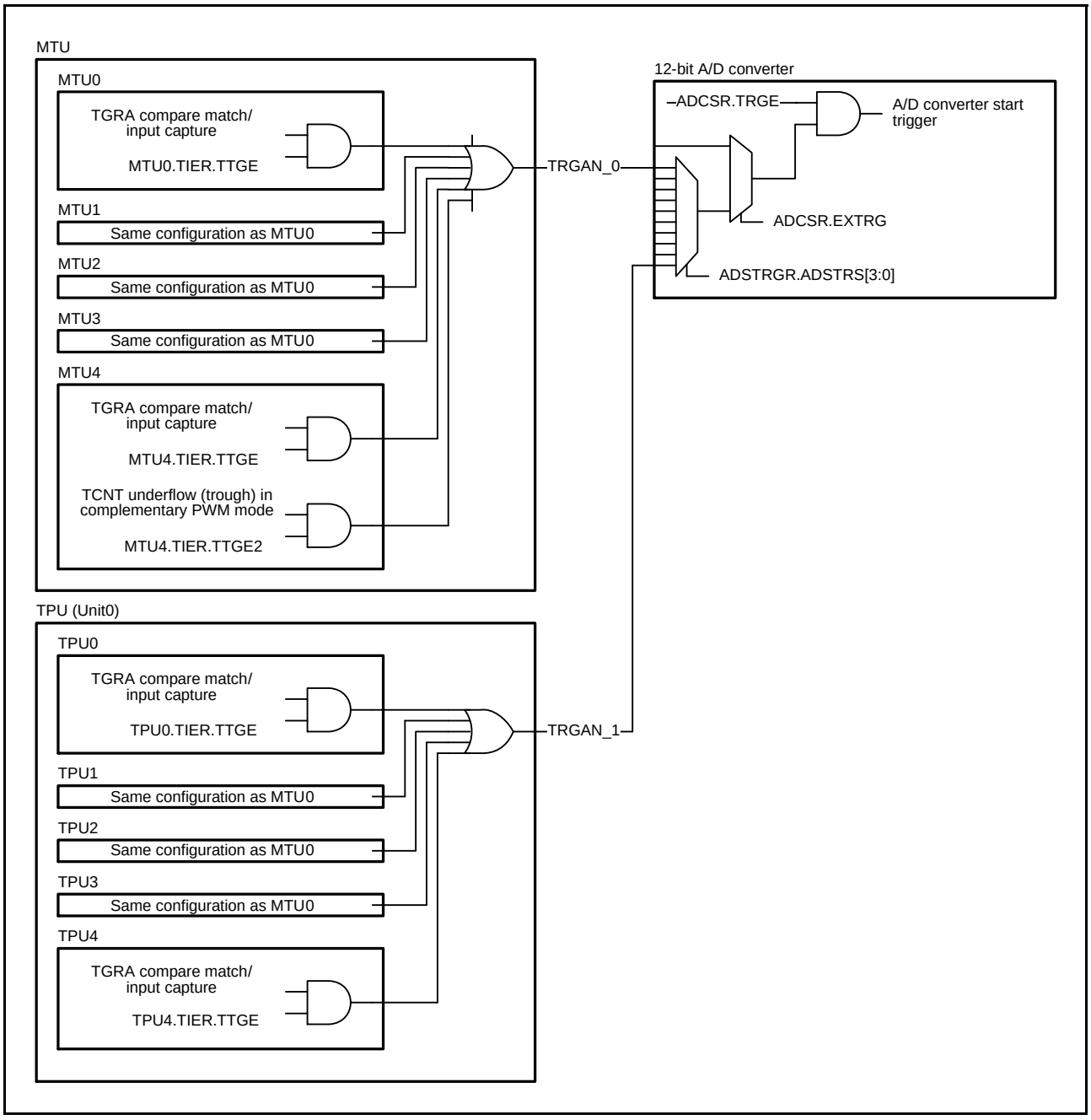


Figure 40.8 Connection of MTU TRGAN_0 Output and TPU (unit 0) TRGAN_1 Output to A/D Converter

40.3.8.3 Starting A/D Conversion Using MTU TRG0EN_0 and TRG0FN_0

Upon the TGRE compare match of the MTU0, the trigger signal TRG0EN_0 can be generated and A/D conversion can be started with TRG0EN_0. Similarly, upon the TGRF compare match of the MTU0, the trigger signal TRG0FN_0 can be generated and A/D conversion can be started.

Figure 40.9 shows the connection of MTU TRG0EN_0 output and TRG0FN_0 output to the A/D converter.

To start A/D conversion upon the TGRE compare match of the MTU0, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, and set ADSTRGR.ADSTRS[3:0] to 0101b (source: TRG5N; corresponding trigger: TRG0EN_0).

To start A/D conversion upon the TGRF compare match of the MTU0, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, and set ADSTRGR.ADSTRS[3:0] to 0110b (source: TRG6N; corresponding trigger: TRG0FN_0).

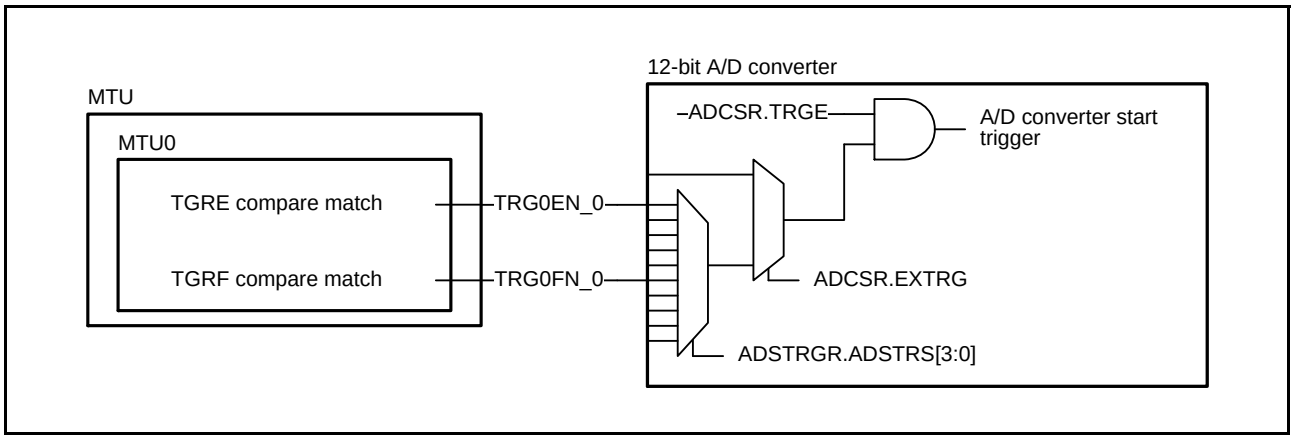


Figure 40.9 Connection of MTU TRG0EN_0 Output and TRG0FN_0 Output to A/D Converter

40.3.8.4 Starting A/D Conversion Using MTU TRG4ABN_0 and TPU TRG4ABN_1

Upon the compare match of the MTU4 generated using the delayed A/D conversion start request function, the trigger signal TRG4ABN_0 can be generated and A/D conversion can be started. Similarly, upon the TRGA input capture/ compare match of the TPU0 in the TPU (unit 0), the trigger signal TRG4ABN_1 can be generated and A/D conversion can be started.

Figure 40.10 shows the connection of MTU TRG4ABN_0 output and TPU (unit 0) TRG4ABN_1 output to the A/D converter. To use the delayed A/D conversion start request function of the MTU4 to start A/D conversion upon the compare match between the TADCORA and TCNT countup, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, set ADSTRGR.ADSTRS[3:0] to 0111b (source: TRG7N; corresponding trigger: TRG4ABN_0), set the cycle to MTU4.TADCOBRA/MTU4.TADCOBRB and MTU4.TADCORA/MTU4.TADCORB, and set MTU4.TADCR.UT4AE to 1.

Also, when A/D conversion is to be started by input capture to or compare match with TGRA in TPU0, set the ADCSR.TRGE bit to 1, the ADCSR.EXTRG bit to 0, the ADSTRGR.ADSTRS[3:0] bits to 1000b (selecting TRG8N as the source and TRG4ABN_1 as the corresponding trigger), and the TPU0.TIER.TTGE bit to 1.

For details of the delayed A/D conversion start request function, see section 22.3.9, A/D Converter Start Request Delaying Function.

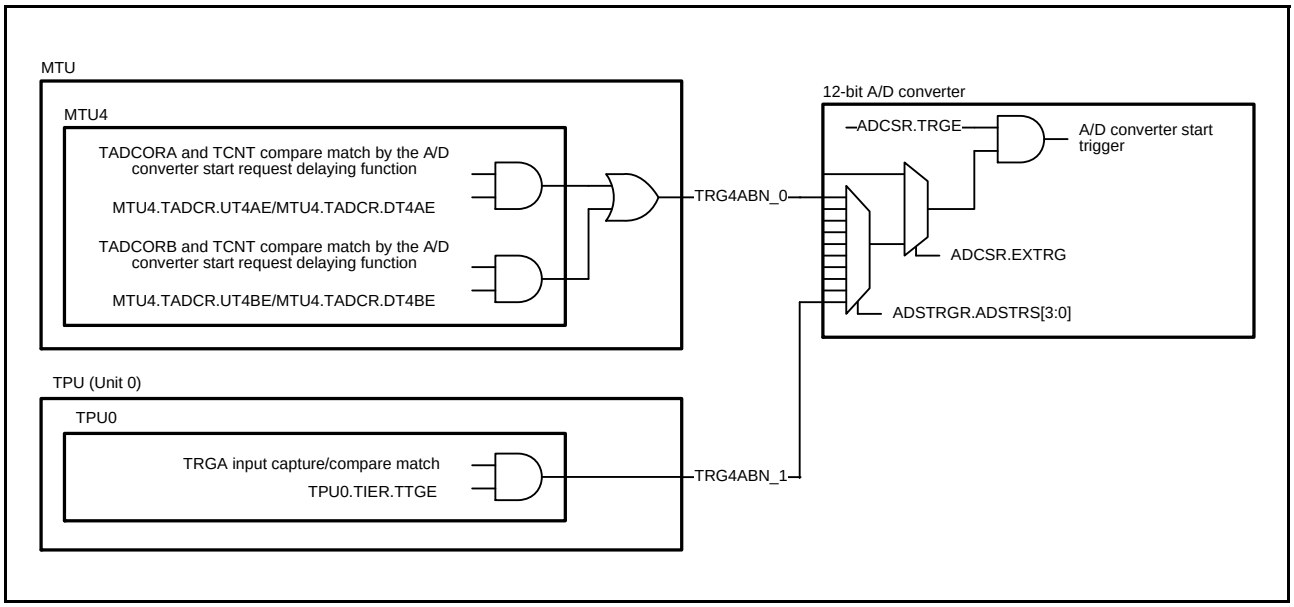


Figure 40.10 Connection of MTU TRG4ABN_0 Output and TPU (Unit 0) TRG4ABN_1 Output to A/D Converter

40.3.8.5 Starting A/D Conversion Using TMR TMTRG0AN_0 and TMTRG0AN_1

Upon the TCORA compare match (compare match A) of the TMR0 in the TMR (unit 0), A/D conversion can be started. Similarly, upon the TCORA compare match (compare match A) of the TMR2 in the TMR (unit 1), A/D conversion can be started.

Figure 40.11 shows the connection of TMR (unit 0 and unit 1) TMTRG0AN_0 output and TMTRG0AN_1 output to the A/D converter.

To start A/D conversion upon the TCORA compare match (compare match A) of the TMR0 in the TMR (unit 0), set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, set ADSTRGR.ADSTRS[3:0] to 1001b (source: TRG9N; corresponding trigger: TMTRG0AN_0), and set TMR0.TCSR.ADTE to 1.

To start A/D conversion upon the TCORA compare match (compare match A) of the TMR2 in the TMR (unit 1), set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, set ADSTRGR.ADSTRS[3:0] to 1010b (source: TRG10N; corresponding trigger: TMTRG0AN_1), and set TMR2.TCSR.ADTE to 1.

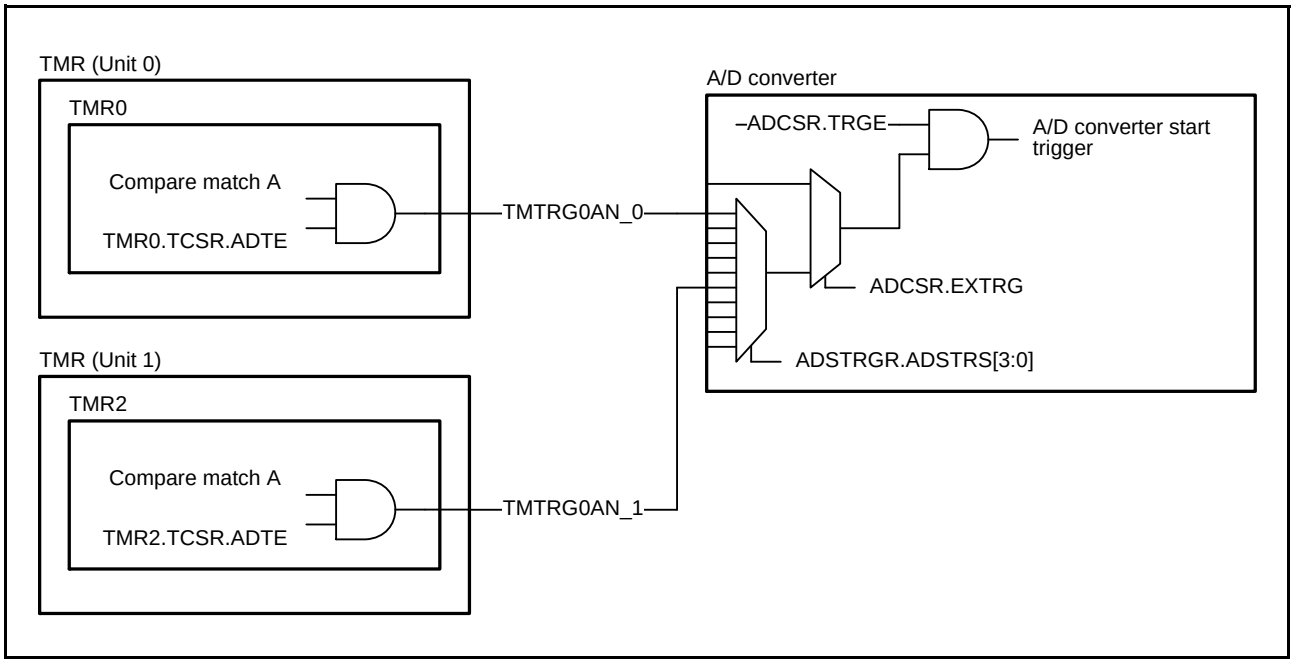


Figure 40.11 Connection of TMR (Unit 0 and Unit 1) TMTRG0AN_0 Output and TMTRG0AN_1 Output to A/D Converter

40.4 Interrupt Sources and DMA Transfer Requests

40.4.1 Interrupt Request on Completion of Each Scanning Conversion

The 12-bit A/D converter can send a scan end interrupt request S12ADI0 interrupt to the CPU.

By setting the ADIE bit in ADCSR to 1, an S12ADI0 interrupt is enabled; by clearing the ADIE bit to 0, an S12ADI0 interrupt is disabled.

In addition, the DTC or DMAC can be started up when an S12ADI0 interrupt is generated. Using an S12ADI0 interrupt to allow the DMAC to read the converted data enables continuous conversion without burden on software.

For details on DTC settings, see section 19, Data Transfer Controller (DTCa), and for details on DMAC settings, see section 17, DMA Controller (DMACA).

40.5 Usage Notes

40.5.1 Notes on Reading Data Register

The A/D data registers, A/D temperature sensor data register, and A/D internal reference voltage data register should be read in the units of word. If read in the units of byte that is classified into upper byte and the lower byte, the A/D converted values read in the units of upper byte are likely to vary from the A/D converted values read in the units of lower byte. Thus, the reading should not be carried out in the units of byte.

40.5.2 Notes on Forcibly Stopping A/D Conversion

When an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, an unintended A/D conversion may start by just setting ADCSR.ADST bit to 0 to stop A/D conversion forcibly. Therefore, to make the 12-bit A/D convertor stop for sure, set the ADCSR.TRGE bit to 0 to select the software trigger as the trigger to start A/D conversion, and then set the ADCSR.ADST bit to 0 (to stop A/D conversion).

40.5.3 Notes on Restarting A/D Conversion after a Forcible Stop

Due to the analog section of the converter, although setting the ADCSR.ADST bit to 1 restarts operation of the analog section of the A/D converter, this requires three cycles of the ADCLK. Although the analog section of the 12-bit A/D converter stops when the ADCSR.ADST bit is set to 0 to forcibly stop A/D conversion, this requires time equivalent to two cycles of the ADCLK.

40.5.4 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 stopped. Cancellation of the module stop state makes it possible to access registers. After the cancellation, wait for 10 ms to start A/D conversion. For details, see section 11, Low Power Consumption.

40.5.5 Notes on Entering Low Power Consumption States

When entering module stop mode or software standby mode, make sure to stop the A/D conversion. To do so, set the ADST bit in ADCSR to 0, and allow time for disabling of the analog input to the 12-bit A/D converter.
Follow the procedure given below to ensure that this time is secured.

1. Set the ADCSR.TRGE bit to 0 (software trigger).
2. Clear the ADCSR.ADST bit to 0.
3. Set the ADCSR.CKS[1:0] bits to 11b (PCLK).
4. After confirming that the A/D converter has been disabled (at least six cycles of the PCLK must be secured as waiting time until A/D conversion is stopped), place the LSI in the module stop state mode or software standby mode.

40.5.6 Notes on Canceling Software Standby Mode

When the software standby mode is canceled, wait until the crystal oscillation stabilization time or PLL-circuit stabilization time has elapsed. Then, wait for another 10 ms to start the A/D conversion. For details, see section 11, Low Power Consumption.

40.5.7 Analog Power Supply Pin Setting Range

If this LSI is used outside the range of the following voltage settings, the reliability of the LSI may be adversely affected.

- Analog Input Voltage Range
  The voltage applied to analog input pin ANn should be in the range VREFL0 ≤ VAN ≤ VREFH0.
- Relationship between power supply pins (AVCC0 – AVSS0, VREFH0 – VREFL0, VCC – VSS)
  Establish the relationship of AVCC0 = VCC and AVSS0 = VSS. As shown in Figure 40.12, connect a 0.1-μF capacitor between each of the power supplies to form a closed loop in the shortest way so that AVCC0 = VCC and VREFL0 = AVSS0 = VSS in the source side. When the A/D converter is not used, set VREFH0 = AVCC0 = VCC and VREFL0 = AVSS0 = VSS.
- Range of VREFH0 Pin Settings
  The setting of the reference voltage using the VREFH0 pin should be in the range of VREFH0 ≤ AVCC0.

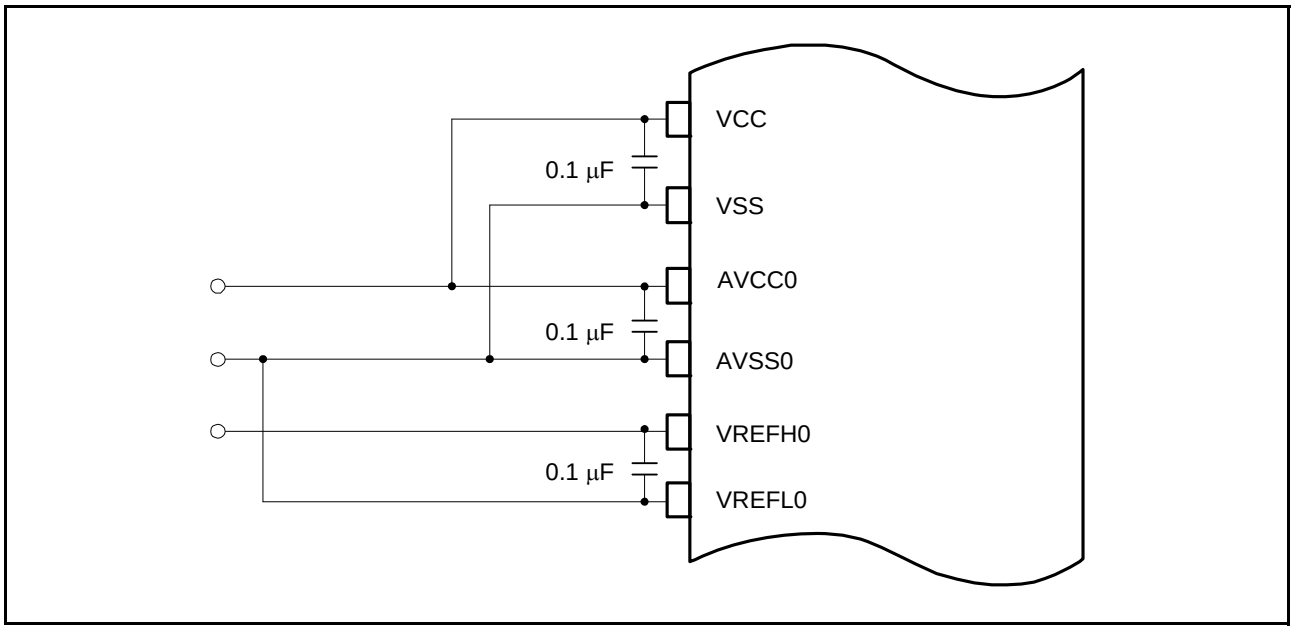


Figure 40.12 Connection Example of Power Supply Pins

40.5.8 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and the layout in which the digital circuit signal lines and analog circuit signal lines cross or are in close proximity to each other should be avoided as much as possible. Failure to do so may result in the incorrect operation of the analog circuitry due to inductance, adversely affecting the A/D conversion values.
In addition, digital circuitry must be isolated from the analog input signals (AN000 to AN020), analog reference voltages (VREFH0, VREFL0), analog power supply (AVCC0), and analog ground (AVSS0). AVSS0 should be connected at one point to a stable digital ground (VSS) on the board.

40.5.9 Notes on Noise Countermeasures

To prevent damage due to an abnormal voltage, such as an excessive surge at the analog input pins (AN000 to AN020), connect capacitors between the AVCC0 and AVSS0, and VREFH0 and VREFL0, and a protection circuit relative to the analog input pins (AN000 to AN020), as shown in Figure 40.13.

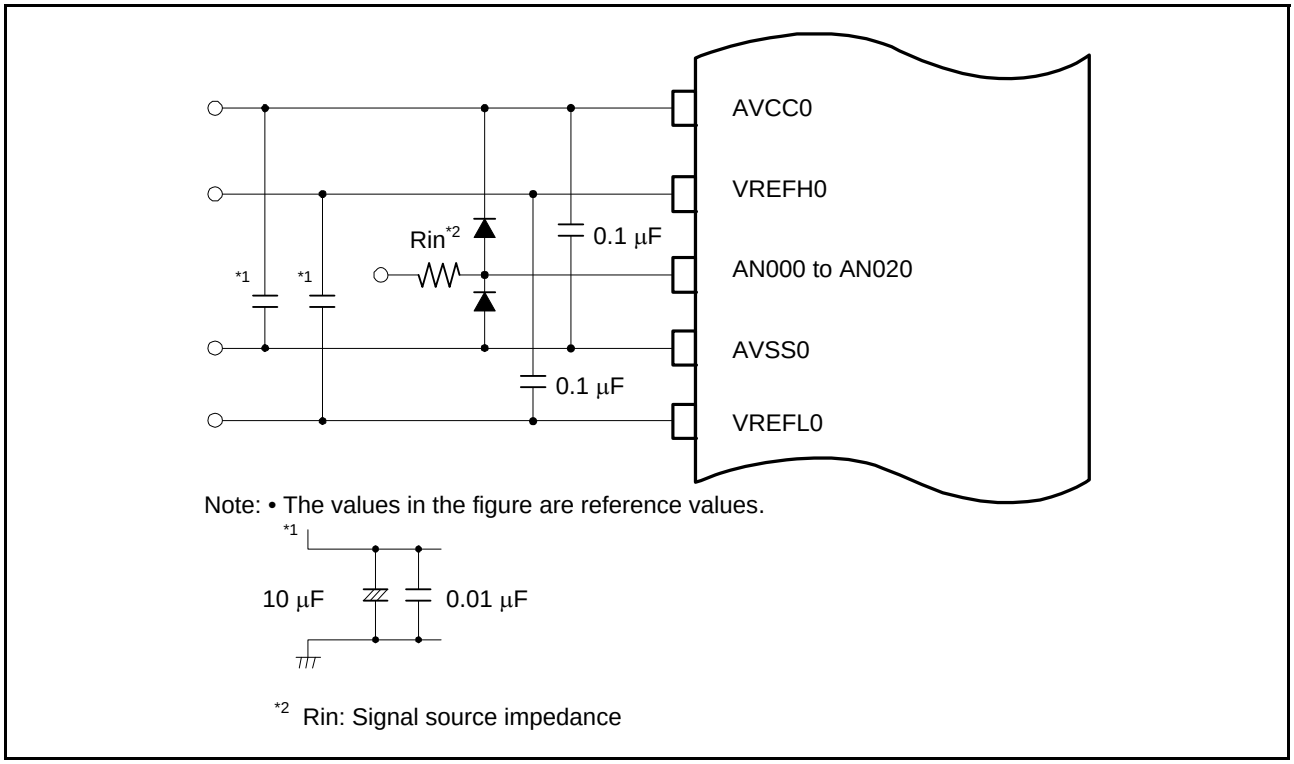


Figure 40.13 Example of Analog Input Pin Protection Circuit

40.5.10 Port Setting for Using 12-bit A/D Converter Inputs

If at least one pin from among the pins of ports 0, 4, 9, and D is used as analog input to the 12-bit A/D converter, ports 0, 4, 9, and D should not be used for port output.

41. 10-Bit A/D Converter (ADb)

41.1 Overview

The RX63N/RX631 Group device includes a single unit of successive approximation type 10-bit A/D converter. The unit allows
up to 8 analog input channels (AN0 to AN7) and one extended analog input channel (ANEX1) to be selected.
The 10-bit A/D converter has two operating modes: single channel mode which converts the analog input of the specified single channel or a single extended analog input for only once and scan mode which continuously converts the analog inputs of the specified channels up to eight or the extended analog input 1 (ANEX1).
Table 41.1 lists the specifications of the 10-bit A/D converter and the Table 41.2 lists the function overview of the 10-bit A/D converter. Figure 41.1 shows the block diagram of the 10-bit A/D converter.

Table 41.1 Specifications of 10-Bit A/D Converter

Item	Specifications
Number of units	Single unit
Input channels	8 channels + one extended analog input
A/D conversion method	Successive approximation method
Resolution	10 bits
Conversion time	1.0 µs per 1 channel (when operating peripheral module clock PCLK = 50 MHz)
A/D conversion clock	4 types: PCLK, PCLK/2, PCLK/4, PCLK/8
Operating modes	• Single channel mode: A/D conversion is to be performed for only once on the analog input of the specified single channel or a single extended analog input. • Scan mode Continuous scan mode: A/D conversion is to be performed sequentially on the analog inputs of the specified channels up to eight or the extended analog input 1 (ANEX1). Single scan mode: A/D conversion is to be performed for one cycle on the analog inputs of the specified channels up to eight or the extended analog input 1 (ANEX1).
Conditions of A/D conversion start	• Software trigger • 6 synchronous A/D conversion triggers from MTU, TPU, or TMR • 1 asynchronous A/D conversion trigger A/D conversion can be started by the ADTRG# pin.
Function	• Sample-and-hold function • Number of sampling states is adjustable. • Self-diagnostic functions of the 10-bit A/D converter
Interrupt source	• After the end of A/D conversion, an interrupt request (ADI0) is generated. • DMAC and DTC can be started by an ADI0 interrupt.
Low-power consumption function	Module stop state can be set.

Table 41.2 Function Overview of the 10-Bit A/D Converter

Item			Function/Internal Trigger Source	
Analog input channel			AN0 to AN7, ANEX1	
A/D conversion start conditions	Software	Software trigger	Enabled	
	Asynchronous trigger	ADTRG#	Enabled	
	Synchronous trigger (MTU, TMR, TPU) *1	TRG0AN_0	MTU0.TGRA and MTU0.TCNT	Input-capture/compare-match
		TRGAN_0	MTU0.TGRA and MTU0.TCNT	Input-capture/compare-match
			MTU1.TGRA and MTU1.TCNT	
			MTU2.TGRA and MTU2.TCNT	
			MTU3.TGRA and MTU3.TCNT	
			MTU4.TGRA and MTU4.TCNT	
			MTU4.TCNT	TCNT underflow (trough) in complementary PWM mode
		TRGAN_1	TPU0.TGRA	Input-capture/compare-match
			TPU1.TGRA	
			TPU2.TGRA	
			TPU3.TGRA	
			TPU4.TGRA	
		TRG4ABN_0	MTU4.TADCORA and MTU4.TCNT or MTU4.TADCORB and MTU4.TCNT	Compare-match using delayed A/D conversion start request function
		TRG4ABN_1	TPU0.TGRA	Input-capture/compare-match
		TMTRG0AN_0	TMR0.TCORA and TMR0.TCNT	Compare-match
Interrupt			ADI0 interrupt	
Module stop function setting*2			MSTPCRA.MSTPA23 bit	

Note 1. 0 and 1 added to synchronous trigger indicate unit number. For synchronous trigger output setting, see section 22.4.3, A/D Converter Activation, and section 26.6.2, A/D Converter Activation.
Note 2. For details, see section 11, Low Power Consumption.

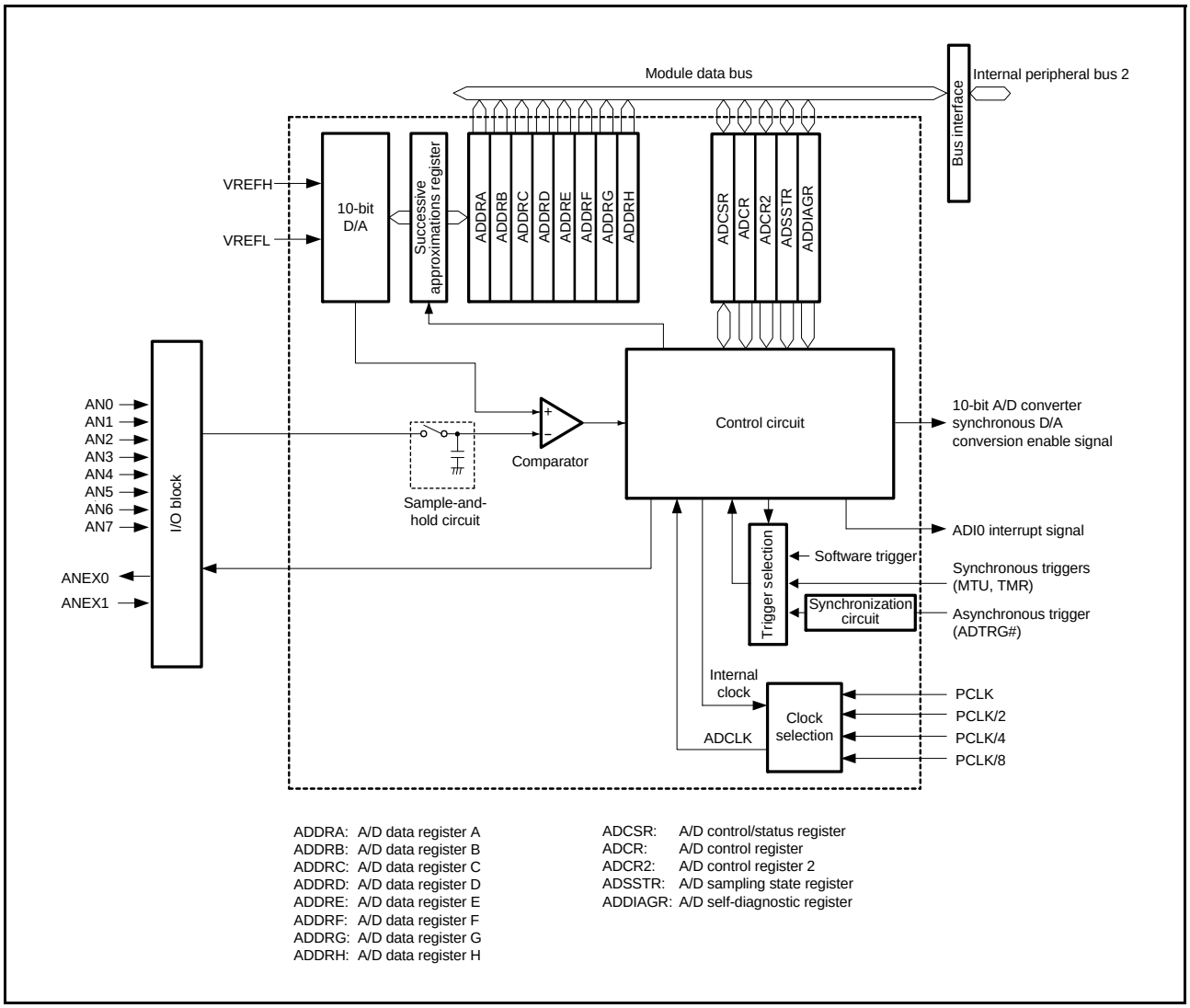


Figure 41.1 Block Diagram of 10-Bit A/D Converter

Table 41.3 indicates the input pins of the 10-bit A/D converter.

Table 41.3 Input Pins of 10-Bit A/D Converter

Pin Name	Input/Output	Function
AN0 to AN7	Input	Analog input pins
ANEX1	Input	Extended analog input pin
ANEX0	Output	Extended analog output pin
ADTRG#	Input	Asynchronous trigger input pin for starting A/D conversion
VREFH	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This is used as the analog power supply for each of the modules. Connect this pin to VCC if neither 10-bit A/D converter nor D/A converter are used.
VREFL	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This is used as the analog ground for each of the modules. Set this pin to the same potential as the VSS pin.

41.2 Register Descriptions

41.2.1 A/D Data Register y (ADDRy) (y = A to H)

Address(es): ADDRA 0008 9800h, ADDRB 0008 9802h, ADDRC 0008 9804h, ADDRD 0008 9806h
ADDRE 0008 9808h, ADDRF 0008 980Ah, ADDRG 0008 980Ch, ADDRH 0008 980Eh

• ADCSR2.DPSEL bit = 0 (Flush-right)

	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

• ADCSR2.DPSEL bit = 1 (Flush-left)

	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

ADDRy are 16-bit read-only registers, which store an A/D conversion result.
Table 41.4 lists the analog input channels and corresponding ADDRy registers, and Table 41.5 lists the extended analog input 1 (ANEX1) and corresponding ADDRy registers.
10-bit data can be relocated by setting the ADCR2.DPSEL bit.
Bits "—" are read as 0. The write value should be 0.

Table 41.4 Analog Input Channels and Corresponding ADDRy Registers

Analog Input Channel	ADDRy Register
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

Table 41.5 Extended Analog Input 1 (ANEX1) and Corresponding ADDRy Registers

Source Analog Input Channel	A/D Conversion Target	ADDRy Register
AN0	ANEX1	ADDRA
AN1		ADDRB
AN2		ADDRC
AN3		ADDRD
AN4		ADDRE
AN5		ADDRF
AN6		ADDRG
AN7		ADDRH

41.2.2 A/D Control/Status Register (ADCSR)

Address(es): 0008 9810h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADIE	ADST	—	—	CH[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description		R/W
b2 to b0	CH[2:0]	Channel Select*1	When ADCR.MODE[1:0] = 00b b2 b0 0 0 0: AN0 0 0 1: AN1 0 1 0: AN2 0 1 1: AN3 1 0 0: AN4 1 0 1: AN5 1 1 0: AN6 1 1 1: AN7	When ADCR.MODE[1:0] = 10b or 11b b2 b0 0 0 0: AN0 0 0 1: AN0, AN1 0 1 0: AN0 to AN2 0 1 1: AN0 to AN3 1 0 0: AN0 to AN4 1 0 1: AN0 to AN5 1 1 0: AN0 to AN6 1 1 1: AN0 to AN7	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.		R/W
b5	ADST	A/D Start	0: Stops A/D conversion 1: Starts A/D conversion		R/W
b6	ADIE	A/D Interrupt Enable	0: ADI0 interrupt is disabled by completing A/D conversion 1: ADI0 interrupt is enabled by completing A/D conversion		R/W
b7	—	Reserved	The read value is undefined. The write value should be 0.		R/W

Note 1. The Pn.PDR.Bi bit for the analog input should be set to 0 (input port) and the Pn.PMR.Bi bit should be set to 0 (disabling the input buffer and fixing the input signal to the high level) (n = D, E, i = 7 to 0). For details, see section 20, I/O Ports and section 21, Multi-Function Pin Controller (MPC).

Set the CH[2:0] and ADIE bits while the ADST bit is 0 (A/D conversion stopped).

CH[2:0] Bits (Channel Select)

These bits select analog input channels that allow A/D conversion.

- Single channel mode (ADCR.MODE[1:0] bits = 00b)
  Select the single analog input channel that allows A/D conversion.
- Scan mode (ADCR.MODE[1:0] bits = 10b or 11b)
  Select up to 8 analog input channels that allow A/D conversion.

ADST Bit (A/D Start)

The ADST bit starts/stops A/D conversion.

Before setting the ADST bit to 1, complete the setting for A/D conversion clock and the operation mode.

[Setting conditions]

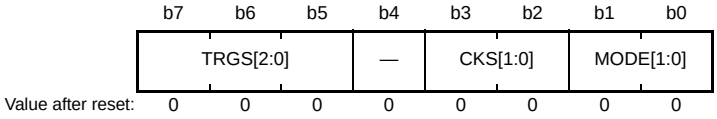
- When 1 is written by software
- Detection of the trigger selected by the.TRGS[2:0] bits in ADCR

[Clearing conditions]

- When 0 is written by software
- The A/D conversion is completed in single channel mode
- The A/D conversion is completed on every selected channel in single scan mode.

41.2.3 A/D Control Register (ADCR)

Address(es): 0008 9811h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	MODE[1:0]	Operation Mode Select	b1 b0 0 0: Single channel mode 0 1: Setting prohibited 1 0: Continuous scan mode 1 1: Single scan mode	R/W
b3, b2	CKS[1:0]	Clock Select	b3 b2 0 0: PCLK/8 0 1: PCLK/4 1 0: PCLK/2 1 1: PCLK	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b5	TRGS[2:0]	Trigger Select	b7 b5 0 0 0: Software trigger 0 0 1: Compare-match/input-capture A from MTU0 to MTU4 0 1 0: Compare-match from TMR0 0 1 1: Trigger from ADTRG# 1 0 0: Compare-match/input-capture A from MTU0 1 0 1: Compare-match/input-capture A from TPU0 to TPU4 1 1 0: Compare-match from MTU4 1 1 1: Compare-match input-capture A from TPU0	R/W

Set the MODE[1:0] and CKS[1:0] bits while the ADST bit in ADCSR is 0 (A/D conversion stopped).

CKS[1:0] Bits (Clock Select)

These bits set the frequency of the A/D conversion clock (ADCLK) and thus select the A/D conversion time.
Set the frequency of ADCLK above 4 MHz.
For details, see section 41.3.4, Input Sampling and A/D Conversion Time.

41.2.4 A/D Control Register 2 (ADCR2)

Address(es): 0008 9812h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSEL	EXOEN	EXSEL[1:0]		—	—	—	—
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 read as 0. The write value should be 0.	R/W
b5, b4	EXSEL[1:0]	Extended Analog Input Select	b5 b4 0 0: Analog input channel (ANxx) 0 1: ANEX1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b6	EXOEN	Extended Analog Output Control	0: Output is disabled. 1: Output is enabled.	R/W
b7	DPSEL	ADDRy Format Select	0: A/D data is flush-right. 1: A/D data is flush-left.	R/W

Set the EXSEL[1:0], and EXOEN bits while the ADST bit in ADCSR is 0 (A/D conversion stopped).

EXSEL[1:0] Bits (Extended Analog Input Select)

These bits can select extended analog inputs ANEX1 other than the analog input channel (ANxx).
ANEX1: ANEX0 should be input through the external operational amplifier.
For details, see section 41.3.3, Extended Analog Input.

EXOEN Bit (Extended Analog Output Control)

This bit can control extended analog output (ANEX0).
When output is enabled, an ANxx-multiplexed value is output to ANEX0.
When the EXSEL[1:0] bits are 00b, output should be disabled.

41.2.5 A/D Sampling State Register (ADSSTR)

Address(es): 0008 9813h

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

ADSSTR is an 8-bit readable/writable register that is used to set the sampling time for analog inputs.
Sampling time is adjustable when the signal source impedance of analog input is high and the sampling time is insufficient or the speed of peripheral module clock (PCLK) is low.
Set the value of 02h or larger.
Ensure to set this register while the A/D conversion is stopped (the ADST bit in ADCSR is 0).
For details, see section 41.3.4, Input Sampling and A/D Conversion Time.

41.2.6 A/D Self-Diagnostic Register (ADDIAGR)

Address(es): 0008 981Fh

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DIAG[1:0]	Self-Diagnostic	b1 b0 0 0: Self-diagnostic function is off. 0 1: A/D conversion of Vref × 0 voltage value is enabled. 1 0: A/D conversion of Vref × 1/2 voltage value is enabled. 1 1: A/D conversion of Vref × 1 voltage value is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Specify ADDIAGR while the ADST bit in ADCSR is 0 (A/D conversion stopped).

DIAG[1:0] Bits (Self-Diagnostic)

The self-diagnostic function is used to detect faults in the 10-bit A/D converter. A voltage value is selected to convert from three internally generated voltages: Vref × 0, Vref × 1/2, and Vref × 1.
To perform self-diagnostic, select the voltage value with the DIAG[1:0] bits and start the A/D conversion with the setting below.

- Single channel mode (the ADCR.MODE[1:0] bits = 00b)
- Analog input channel AN0 is only enabled (the ADCSR.CH[2:0] bits = 000b, and ADCR2.EXSEL[1:0] bits = 00b)*1
- A/D conversion is started by software. (the ADCR.TRGS[2:0] bits = 000b)

After A/D conversion ends, the conversion result is stored in the A/D data register A. Then, the value in ADDRA is read by software to determine whether the conversion result is within the normal range (normal) or not (error). The execution time of self-diagnosis is the same as that for A/D conversion of one channel.

Note 1. Set AN0 to perform self-diagnosis of 10-bit A/D converter. Though this setting is required to select the data register to store the conversion result, all-analog input (AN0 to AN7) has no effect.

41.3 Operation

The RX63N/RX631 Group device includes a single unit of 10-bit A/D converter.

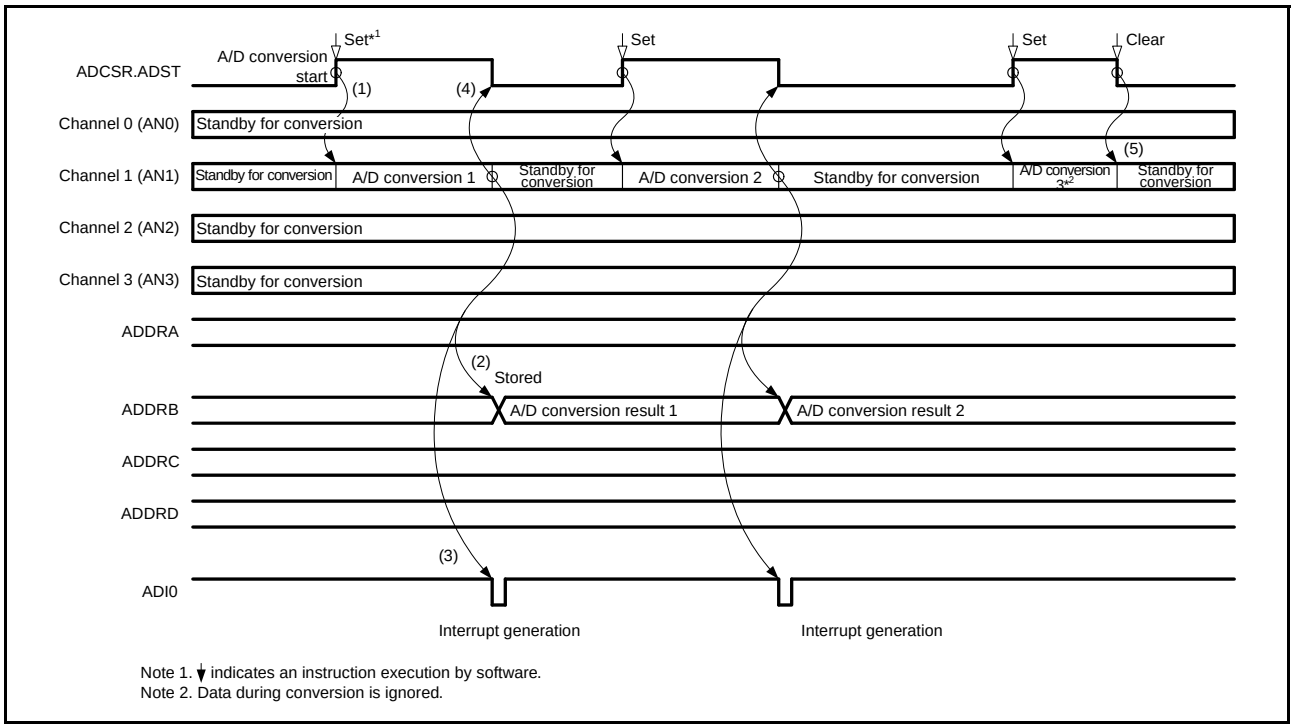
A/D conversion targets can be selected from among 8 analog input channels (AN0 to AN7) and one extended analog input (ANEX1), and then can be input.
The analog input channels and the extended analog inputs are selected using the EXSEL[1:0] bits in ADCR2. Each of the analog input channels (AN0 to AN7), the extended analog input 1 (ANEX1) are exclusively operated.
When the analog input channel (00) is selected with the EXSEL[1:0] bits in ADCR2, the extended analog output (ANEX0) must be disabled by clearing the EXOEN bit in ADCR2 to 0. If output is not disabled, the time-shared analog value will be output from ANEX0. Further, due to internal wiring load of the LSI, the A/D conversion will be delayed.
The 10-bit A/D converter has two operating modes: single channel mode and scan mode.
In single channel mode, A/D conversion is performed only once on the analog input of the specified single channel or an extended analog input.
In scan mode, A/D conversion is performed sequentially on the analog inputs of the specified channels or the extended analog input 1 (ANEX1).
Two types of scan mode are provided: continuous san mode where A/D conversion is repeatedly performed and one-cycle scan mode where one cycle of A/D conversion is performed on the specified channels.

41.3.1 Single Channel Mode

In single channel mode, A/D conversion is performed only once on the analog input of the specified single channel or the single extended analog channel as below.

(1) A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, a synchronous trigger input, or an asynchronous trigger input.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy) of the channel.
(3) When A/D conversion is completed, if the ADIE bit in ADCSR is set to 1 (ADI0 interrupt enable by completing A/D conversion), an ADI0 interrupt request is generated.
(4) The ADST bit is held at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion ends. Then the 10-bit A/D converter enters a wait state.
(5) If the ADST bit is cleared to 0 (A/D conversion stop) during A/D conversion, A/D conversion stops and the 10-bit A/D converter enters a wait state.

Figure 41.2 shows an example of operation when AN1 is selected as an analog input.



Note 1. ▼ indicates an instruction execution by software.
Note 2. Data during conversion is ignored.

Figure 41.2 Example of 10-Bit A/D Converter Operation (Single Channel Mode)

41.3.2 Scan Mode

A case when the analog input channel is selected (the EXSEL[1:0] bits in ADCR2 are set to 00b) is described below. For the case with the extended analog input selected, see section 41.3.3, Extended Analog Input.
In scan mode, A/D conversion is performed sequentially on the analog inputs of the channels up to eight that are specified with the CH[2:0] bits in ADCSR. There are two scan modes: continuous scan mode where A/D conversion is repeatedly performed and one-cycle scan mode where A/D conversion is performed on the specified channels for one cycle.

41.3.2.1 Continuous Scan Mode

In continuous scan mode, A/D conversion is performed sequentially on the analog inputs of the specified channels as below.

(1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, a synchronous trigger input, or an asynchronous trigger input, A/D conversion starts from the first channel (AN0) in the specified channel group.
(2) When A/D conversion for each channel is completed, the A/D conversion result is stored into the corresponding A/D data registers (ADDRy).
(3) When A/D conversion of all the selected channels is completed, if the ADIE bit in ADCSR is set to 1 (ADI0 interrupt enabled by completion of A/D conversion), an ADI0 interrupt request is generated. The 10-bit A/D converter starts A/D conversion from the first channel (AN0).
(4) The ADST bit is not cleared automatically, and steps 2 to 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0 (A/D conversion stop), A/D conversion stops and the 10-bit A/D converter enters a wait state.
(5) After the ADST bit in ADCSR is set to 1, A/D converter starts A/D conversion from the first channel (AN0) again.

Figure 41.3 shows an example of A/D conversion when three channels (AN0 to AN2) are selected for analog input.

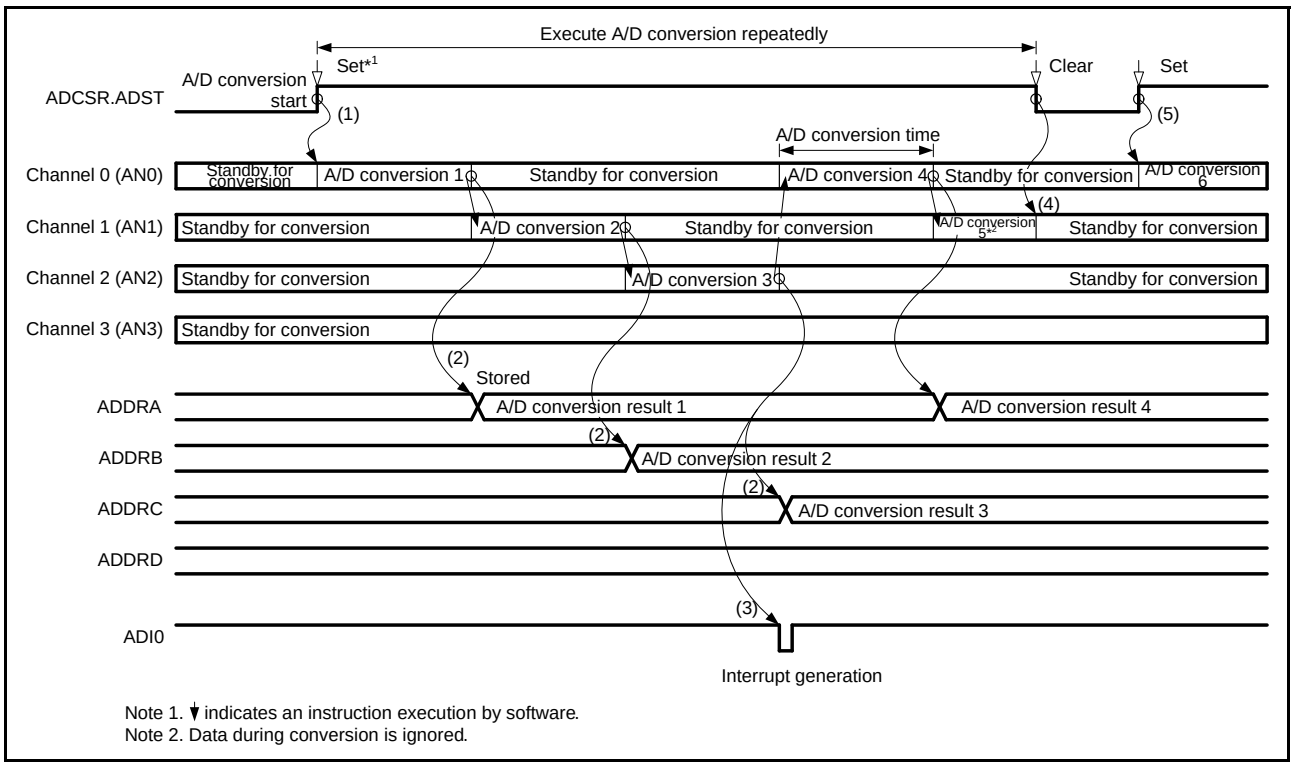


Figure 41.3 Example of 10-Bit A/D Converter Operation (Continuous Scan Mode)

41.3.2.2 Single Scan Mode

In single scan mode, A/D conversion is performed for one cycle on the analog inputs of the specified channels as below.

(1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, a synchronous trigger input, or an asynchronous trigger input, A/D conversion starts from the first channel (AN0) in the specified channel group.
(2) When A/D conversion for each channel is completed, the A/D conversion result is stored into the corresponding A/D data registers (ADDRy).
(3) When A/D conversion is completed for all the selected channels, an ADI0 interrupt request is generated if the ADIE bit in ADCSR is set to 1 (ADI0 interrupt enabled by completion of A/D conversion).
(4) The ADST bit is held at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed, which enters the 10-bit A/D converter into a wait state.

Figure 41.4 shows an example of A/D conversion when three channels (AN0 to AN2) are selected for analog input.

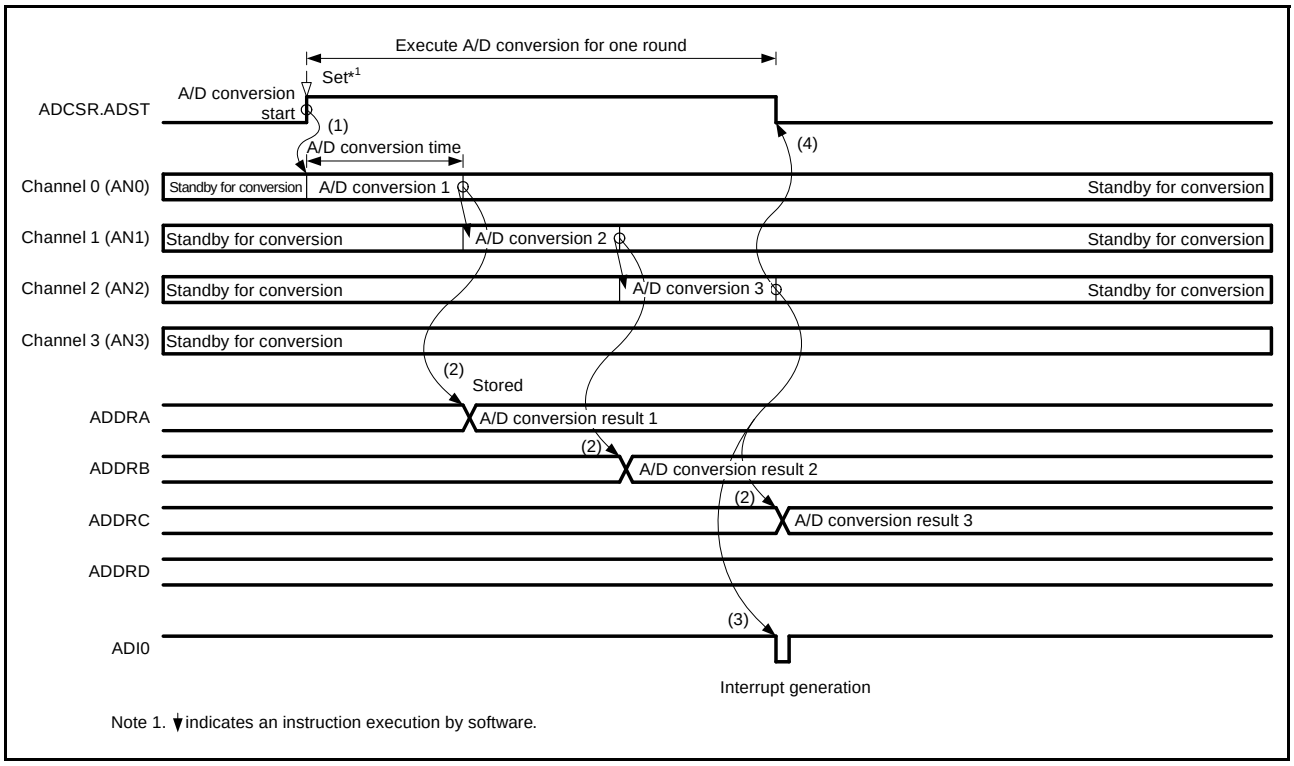


Figure 41.4 Example of 10-Bit A/D Converter Operation (Single Scan Mode)

41.3.3 Extended Analog Input

There is one extended analog input ANEX1. The usage of the extended analog input is determined as given below. Only the following usage method should be used.

The ANEX1 is used when an external operational amplifier is connected to the LSI to perform A/D conversion for the multiple analog values.

Figure 41.5 shows a configuration example of the extended analog input circuit.

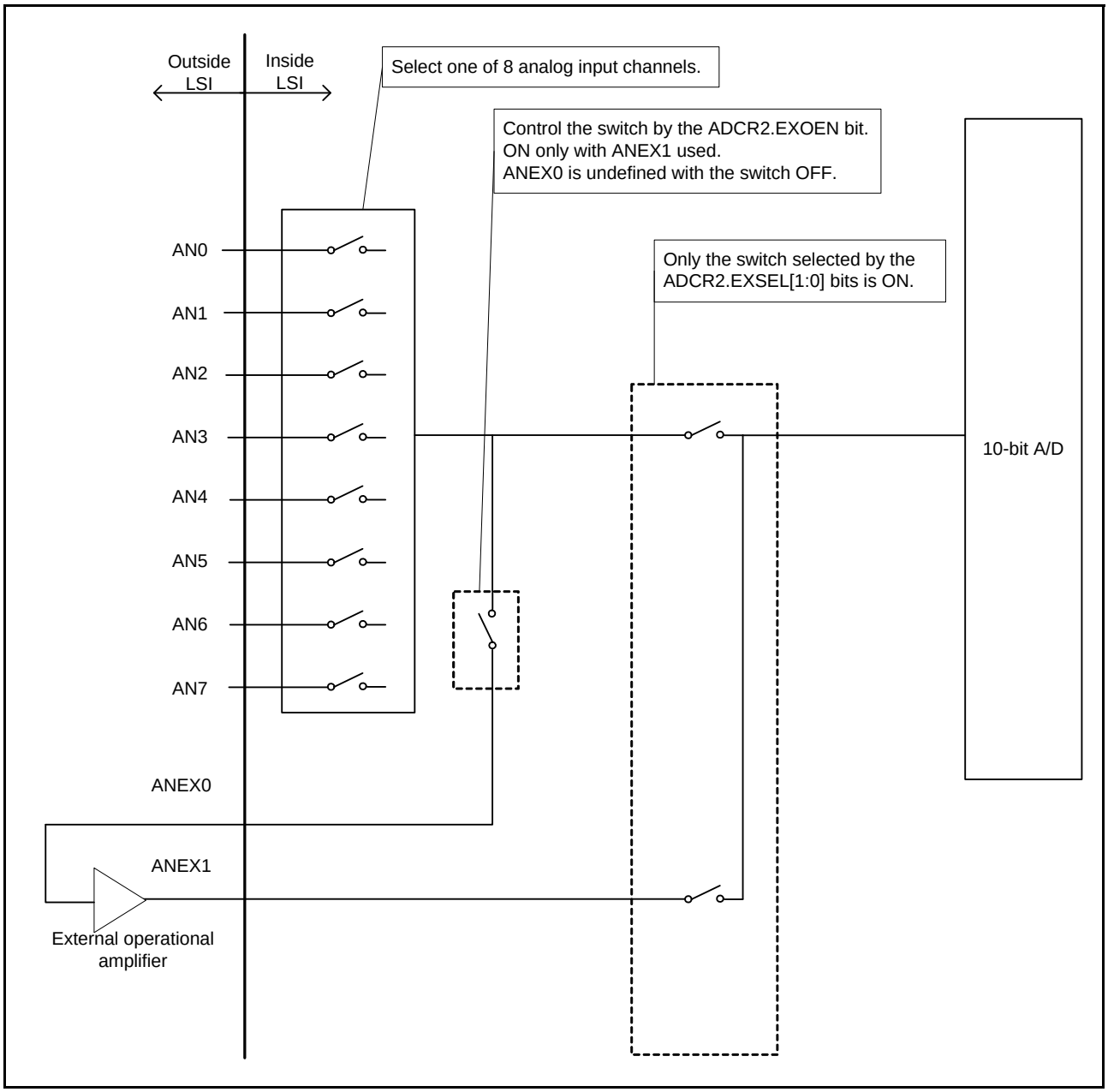


Figure 41.5 Configuration Example of Extended Analog Input Circuit

41.3.3.1 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 (AN0 to AN7). Then the time-shared 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 ADCR2.EXSEL[1:0] bits to 01b. To enable ANEX0 output, set the ADCR2.EXOEN bit to 1. In addition, select single channel mode or scan mode.
Figure 41.6 shows the operation when three channels (AN0, AN1, and AN2) and single scan mode are selected.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, a synchronous trigger input, or an asynchronous trigger input, A/D conversion starts from the first channel (AN0) in the specified channel group.
(2) When A/D conversion is completed, the A/D conversion result is stored in an A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, if the ADCSR.ADIE bit is 1 (ADI0 interrupt enabled by completion of A/D conversion), an ADI0 interrupt request is generated.
(4) The ADCSR.ADST bit is held at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed, and the 10-bit A/D converter enters a wait state.

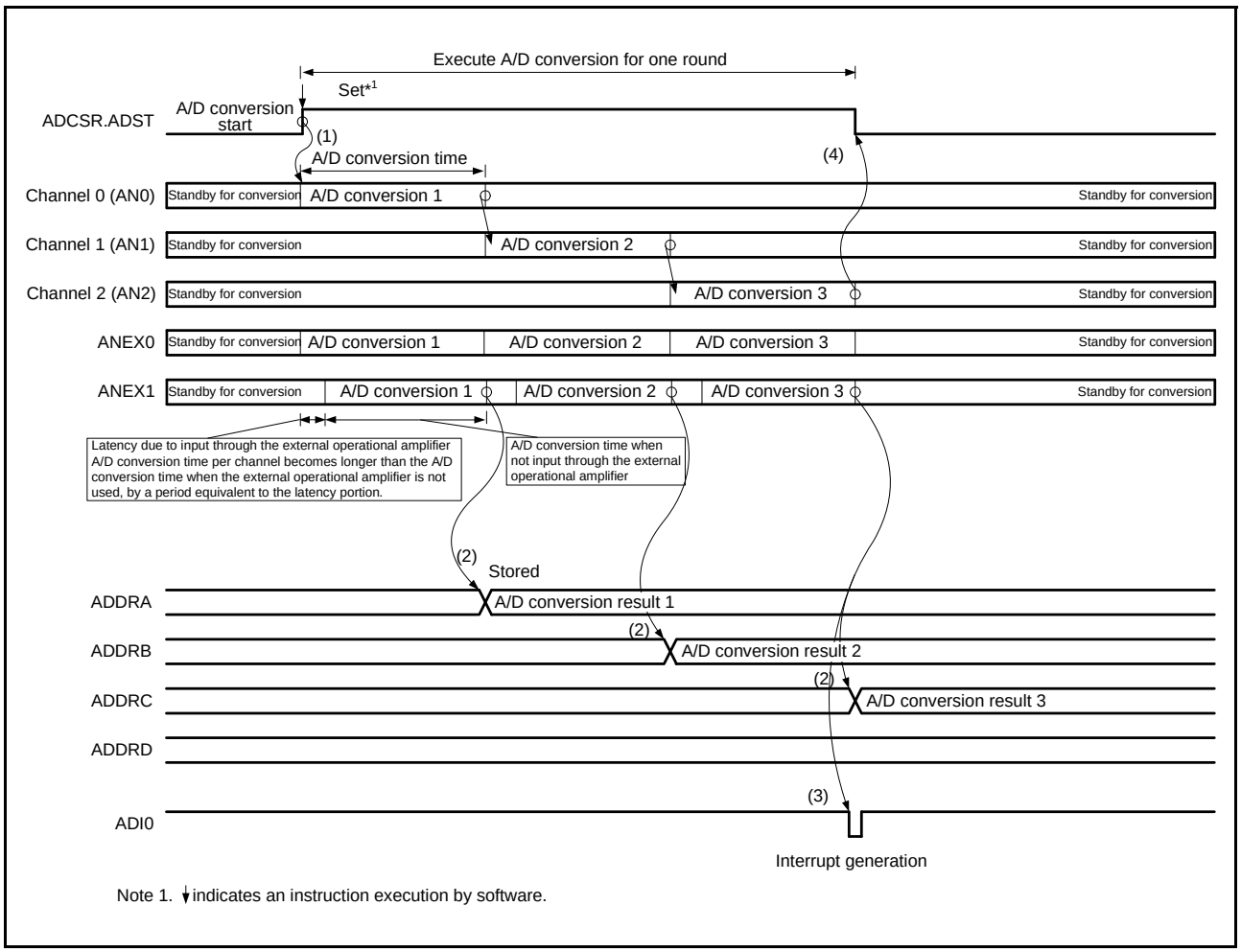


Figure 41.6 ANEX1 Input Operation Example (Single Scan Mode)

When the extended analog input is selected, the individual A/D conversion time becomes longer due to the latency of the operational amplifier, as compared to the A/D conversion that is directly performed for the analog input channels.

41.3.4 Input Sampling and A/D Conversion Time

The 10-bit A/D converter performs sampling of analog input when the A/D conversion start delay time (tD) has elapsed after A/D conversion start conditions are generated by software, a synchronous trigger, or an asynchronous trigger. After sampling, A/D conversion is started.

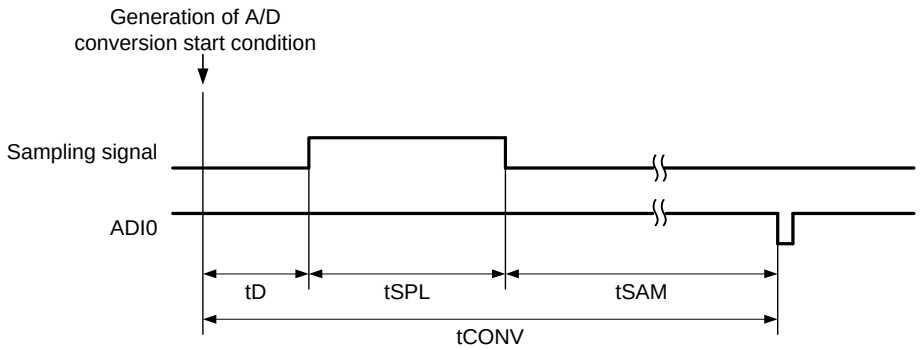
Figure 41.7 shows the A/D conversion timing.

As shown in the figure, the A/D conversion time (tCONV) following the generation of A/D conversion start conditions is a total of the A/D conversion start delay time (tD), the input sampling time (tSPL), and the successive conversion time (tSAM). The subsequent A/D conversion time (tCONV) is addition of tSPL and tSAM.

The sampling time (tSPL) is the time for charging the input capacitance of the 10-bit A/D converter's sample-and-hold circuit. If the impedance of the analog input signal source is too high to ensure sufficient sampling time, or the peripheral module clock (PCLK) is running at low speed, the sampling time can be adjusted by using the ADSSTR.

The successive conversion time (tSAM) is fixed at 25 cycles of ADCLK.

The samples of ADSSTR settings are listed in Table 41.6 and the A/D conversion times in Table 41.7.



tD: A/D conversion start delay time
tSPL: Input sampling time
tSAM: Successive conversion time
tCONV: A/D conversion time

Figure 41.7 A/D Conversion Timing

Table 41.6 Example of ADSSTR Setting

Setting Conditions	Setting Range	Sampling Time*1
Standard (initial value)	19h	0.5 µs (When PCLK = ADCLK = 50 MHz)
Analog input signal source impedance is higher, resulting in insufficient sampling time	1Ah to FFh	Ex: FFh 5.1 µs (When PCLK = ADCLK = 50 MHz)
With ADCLK below 50 MHz, sampling time needs to be less than the initial value.	02h to 18h	Ex: 14h 0.5 µs (When PCLK = ADCLK = 40 MHz)

Note 1. Set the sampling time ≥ 0.5 µs. Sampling time is shown by:

$$\text{Sampling time (µs)} = \frac{\text{Setting value of ADSSTR}}{\text{ADCLK (MHz)}}$$

Table 41.7 A/D Conversion Time

Item	Symbol	Formula	
		Min.	Max.
A/D conversion start delay time (1)	tD	$\frac{3}{\text{PCLK (MHz)}}$	$\frac{1}{\text{ADCLK (MHz)}} + \frac{4}{\text{PCLK (MHz)}}$
Input sampling time (2)	tSPL	$\frac{\text{Setting value of ADSSTR}}{\text{ADCLK (MHz)}}$	
Successive conversion time (3)	tSAM	$\frac{25}{\text{ADCLK (MHz)}}$	
A/D conversion time*1	tCONV	(1) + (2) + (3)	
A/D conversion time*2	tCONV	(2) + (3)	

Note 1. A/D conversion time in single channel mode and scan mode (first round)
Note 2. A/D conversion time in scan mode (after the second round)

The examples of the calculation of A/D conversion times are listed below.

When PCLK = ADCLK = 50 MHz, ADSSTR = 19h, and the conversion is the second round in scan mode,
A/D conversion time (tCONV) = ADSSTR/ADCLK + 25/ADCLK
= 25/50 MHz + 25/50 MHz
= 0.5 µs + 0.5 µs
= 1.0 µs

When PCLK = ADCLK = 40 MHz, ADSSTR = 14h, and conversion is the first (min.) round in scan mode,
A/D conversion time (tCONV) = 3/PCLK + ADSSTR/ADCLK + 25/ADCLK
= 3/40 MHz + 20/40 MHz + 25/40 MHz
= 0.075 µs + 0.5 µs + 0.625 µs
= 1.2 µs

41.3.5 Starting 10-Bit A/D Conversion with Asynchronous Trigger

A/D conversion can be started by the asynchronous trigger (ADTRG# pin) input.
When the ADCR.TRGS[2:0] bits are set to 011b (asynchronous trigger (ADTRG# pin)), the ADCSR.ADST bit is set to 1 (A/D conversion start) on a falling edge of the asynchronous trigger (ADTRG# pin) to start A/D conversion. This timing is shown in Figure 41.8.
Note that when the asynchronous trigger input is already low in using the asynchronous trigger, generation of a falling edge of the internal signal may cause A/D conversion to be activated.

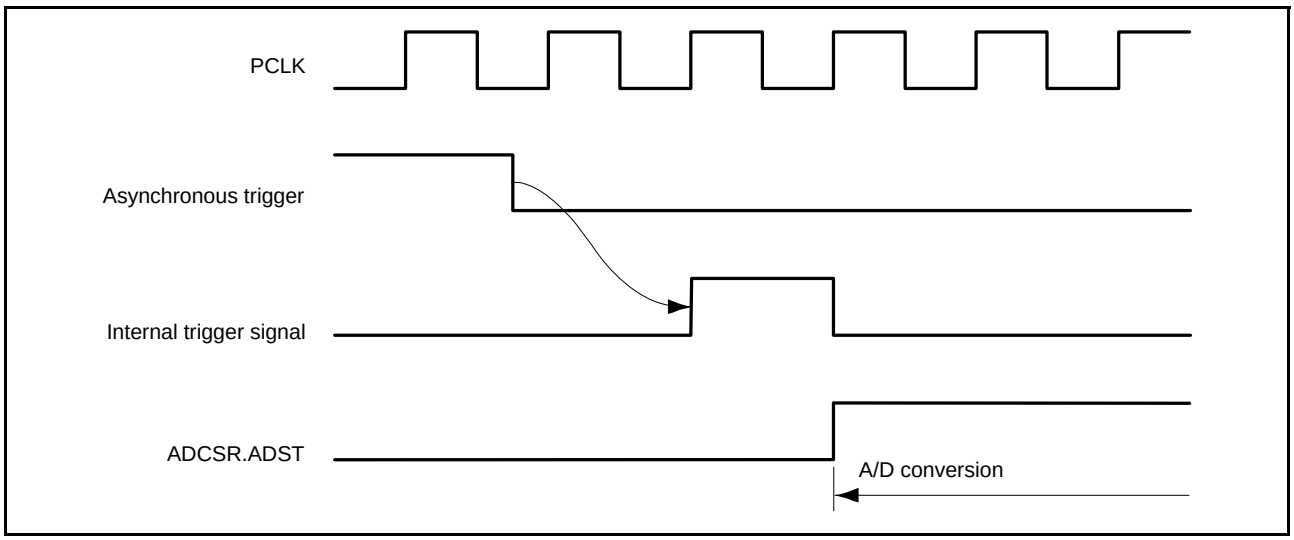


Figure 41.8 Asynchronous Trigger Input Timing

41.3.6 Using Synchronous Triggers from Peripheral Modules to Start A/D Conversion

A/D conversion can be started by synchronous triggers from the MTU, TPU, and TMR modules. When using a synchronous trigger to start A/D conversion, set the ADCR.TRGS[2:0] bits to select the required start source.

41.3.6.1 Starting A/D Conversion Using MTU TRG0AN_0

Upon the TRGA input capture/compare match of the MTU0, the trigger signal TRG0AN_0 can be generated and A/D conversion can be started.
Figure 41.9 shows the connection of MTU TRG0AN_0 output to the A/D converter.
To start A/D conversion upon the TRGA input capture/compare match of the MTU0, set ADCR.TRGS[2:0] in the A/D converter to 100b (trigger signal: TRG0AN_0).

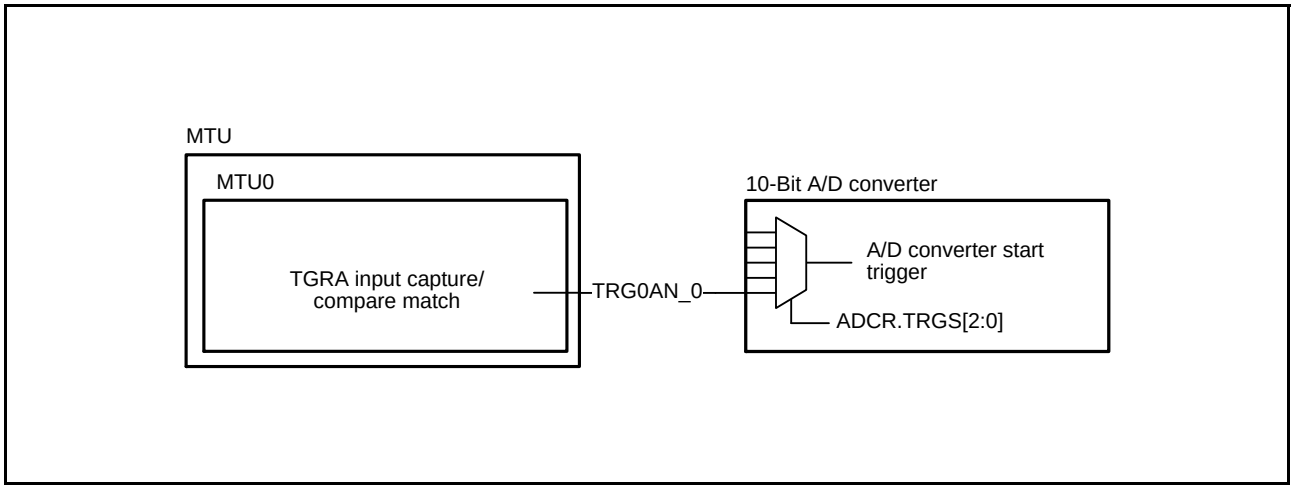


Figure 41.9 Connection of MTU TRG0AN_0 Output to A/D Converter

41.3.6.2 Starting A/D Conversion Using MTU TRGAN_0 and TPU TRGAN_1

Upon the TRGA input capture/compare match of the MTU0 to MTU4 and the TCNT underflow (trough) of the MTU4 in complementary PWM mode, the trigger signal TRGAN_0 can be generated and A/D conversion can be started.
Similarly, upon the TRGA input capture/compare match of the TPU0 to TPU4 in the TPU (unit 0), the trigger signal TRGAN_1 can be generated and A/D conversion can be started.
Figure 41.10 shows the connection of MTU TRGAN_0 output and TPU (unit 0) TRGAN_1 output to the A/D converter.
To start A/D conversion upon the TRGA input capture/compare match of the MTU0 to MTU4 and the TCNT underflow (though) of the MTU4 in complementary PWM mode, set the ADCR.TRGS[2:0] bits to 001b (selecting: TRGAN_0 as the trigger signal), set the MTUn.TIER.TTGE bit (n = 0 to 4), and the MTU4.TIER.TTGE2 bit to 1.
To start A.D conversion upon input capture to or compare match with TGRA in TPU0 to TPU4, set the ADCR.TRGS[2:0] bits to 101b (selecting TRGAN_1 as the trigger signal), and the corresponding TPUn.TIER.TTGE bit (n = 0 to 4) to 1.

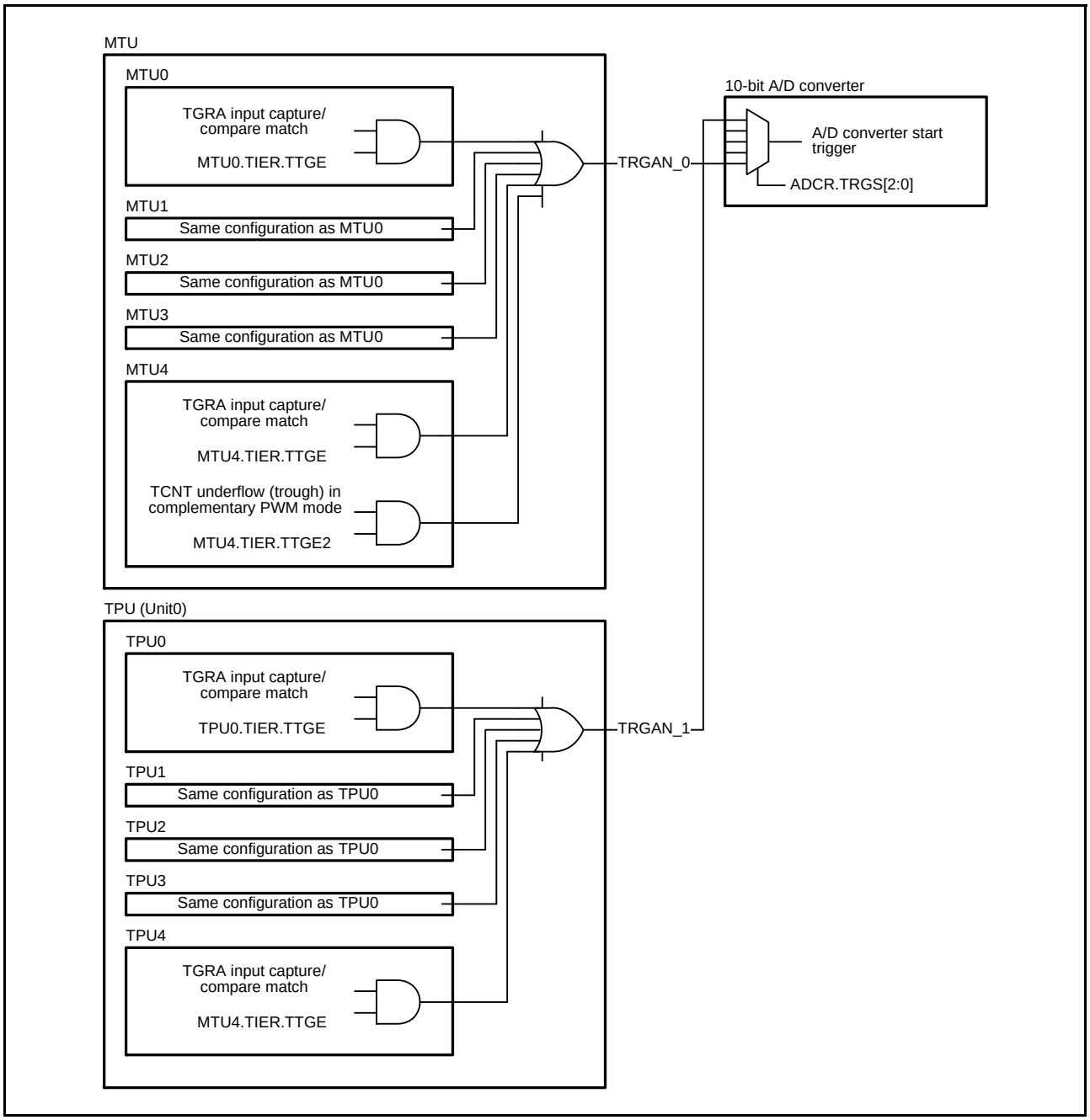


Figure 41.10 Connection of MTU TRGAN_0 Output and TPU (unit 0) TRGAN_1 Output to A/D Converter

41.3.7 Starting A/D Conversion Using MTU TRG4ABN_0 and TPU TRG4ABN_1

Upon the compare match of the MTU4 generated using the delayed A/D conversion start request function, the trigger signal TRG4ABN_0 can be generated and A/D conversion can be started. Similarly, upon the TRGA input capture/ compare match of the TPU0 in the TPU (unit 0), the trigger signal TRG4ABN_1 can be generated and A/D conversion can be started with TRG4ABN_1.

Figure 41.11 shows the connection of MTU TRG4ABN_0 output and TPU (unit 0) TRG4ABN_1 output to the A/D converter.

To use the delayed A/D conversion start request function of the MTU4 to start A/D conversion upon the compare match between the TADCORA and TCNT countup, set ADCR.TRGS[2:0] in the A/D converter to 110b (trigger signal: TRG4ABN_0), set the cycle to MTU4.TADCOBRA/MTU4.TADCOBRB and MTU4.TADCORA/MTU4.TADCORB, and set MTU4.TADCR.UT4AE to 1.

To start A/D conversion upon input capture to or compare match with TGRA in TPU0, set the ADCR.TRGS[2:0] bit to 111b (selecting TRG8N as source and TRG4ABN_1 as the trigger signal) and the TPU0.TIER.TTGE bit to 1.

For details on the delayed A/D conversion start request function, see section 22.3.9, A/D Converter Start Request Delaying Function.

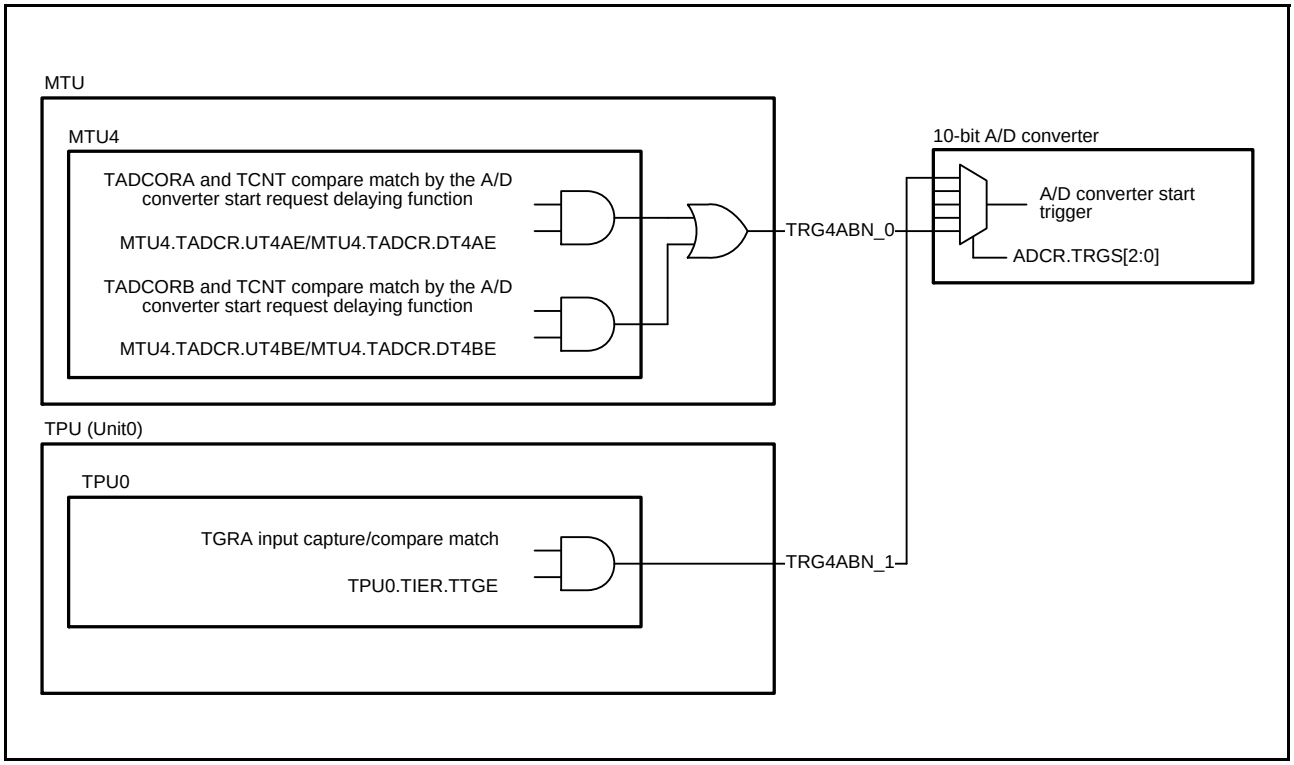


Figure 41.11 Connection of MTU TRG4ABN_0 Output and TPU (unit 0) TRG4ABN_1 Output to A/D Converter

41.3.8 Starting A/D Conversion Using TMR TMTRG0AN_0

Upon the TCORA compare match (compare match A) of the TMR0 in the TMR (unit 0), A/D conversion can be started. Figure 41.12 shows the connection of TMR (unit 0) TMTRG0AN_0 output to the A/D converter.
To start A/D conversion upon the TCORA compare match (compare match A) of the TMR0 in the TMR (unit 0), set ADCR.TRGS[2:0] in the A/D converter to 010b (trigger signal: TMTRG0AN_0), and set TMR0.TCSR.ADTE to 1.

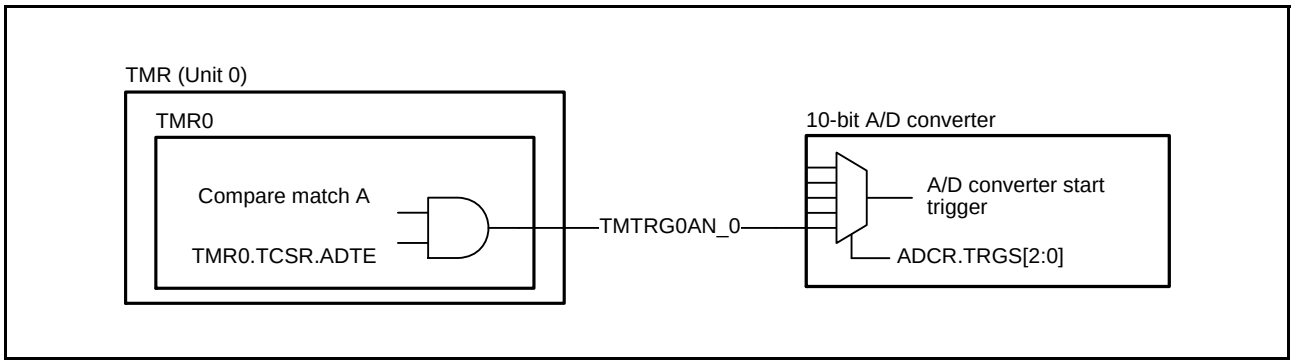


Figure 41.12 Connection of TMR (Unit 0) TMTRG0AN_0 Output to A/D Converter

41.3.9 10-Bit A/D Converter Synchronous D/A Conversion Enable Signal

The 10-bit A/D converter synchronous D/A conversion enable signal output is high when the ADCSR.ADST bit in the 10-bit A/D converter is 0 (waiting for conversion). When the ADCSR.ADST bit is 1 (A/D conversion in progress), a high pulse of one PCLK cycle is output upon completion of A/D conversion. Figure 41.13 shows the output timing of the 10-bit A/D converter synchronous D/A conversion enable signal.

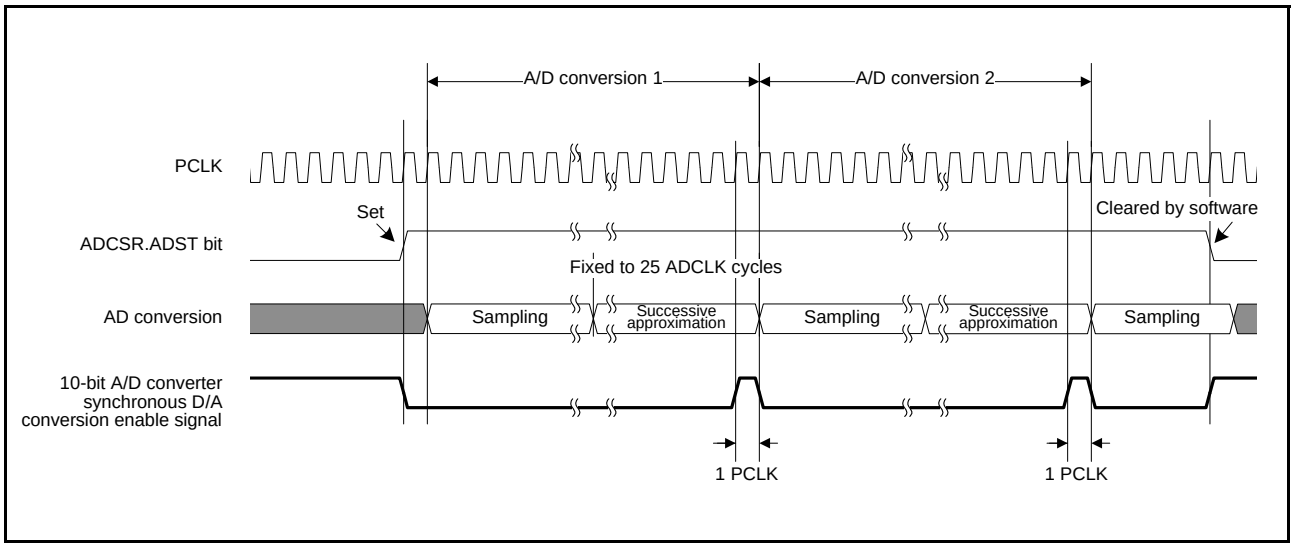


Figure 41.13 Output Timing of the 10-Bit A/D Converter Synchronous D/A Conversion Enable Signal

41.4 Interrupt Sources

The 10-bit A/D converter generates an A/D conversion end interrupt (ADI0) when A/D conversion is terminated with the ADIE bit in ADCSR set to 1 (enabled ADI0 interrupt after A/D conversion completion).
In addition, the DTC or DMAC can be started up when an ADI0 interrupt is generated. Using an ADI0 interrupt to allow the DTC or DMAC to read the converted data enables continuous conversion without burden on software.

Table 41.8 10-Bit A/D Converter Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
ADI0	A/D conversion end	Possible	Possible

41.5 A/D Conversion Accuracy Definitions

The RX63N/RX631 Group's A/D conversion accuracy is defined as below:

- Resolution
  The number of A/D converter digital output codes
- Quantization error
  The deviation inherent in the A/D converter, given by 1/2 LSB (see Figure 41.14)
- Offset error
  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 0000000000b (000h) to 0000000001b (001h) (see Figure 41.15)
- Full-scale error
  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 1111111110b (3FEh) to 1111111111b (3FFh) (see Figure 41.15)
- Nonlinearity error
  The error with respect to the ideal A/D conversion characteristic between zero voltage and the full-scale voltage, excluding offset error, full-scale error, and quantization error (see Figure 41.15).
- Absolute accuracy
  The deviation between the digital value and analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

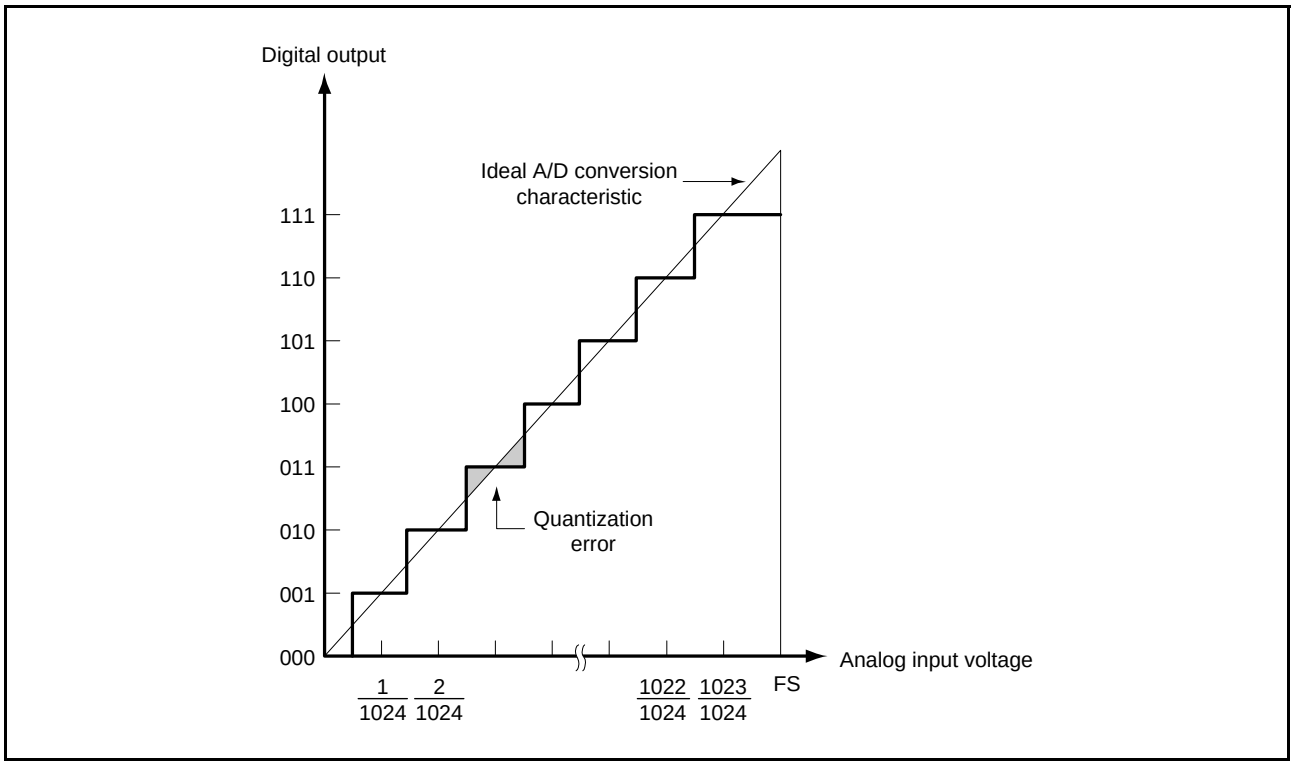


Figure 41.14 A/D Conversion Accuracy Definitions (1)

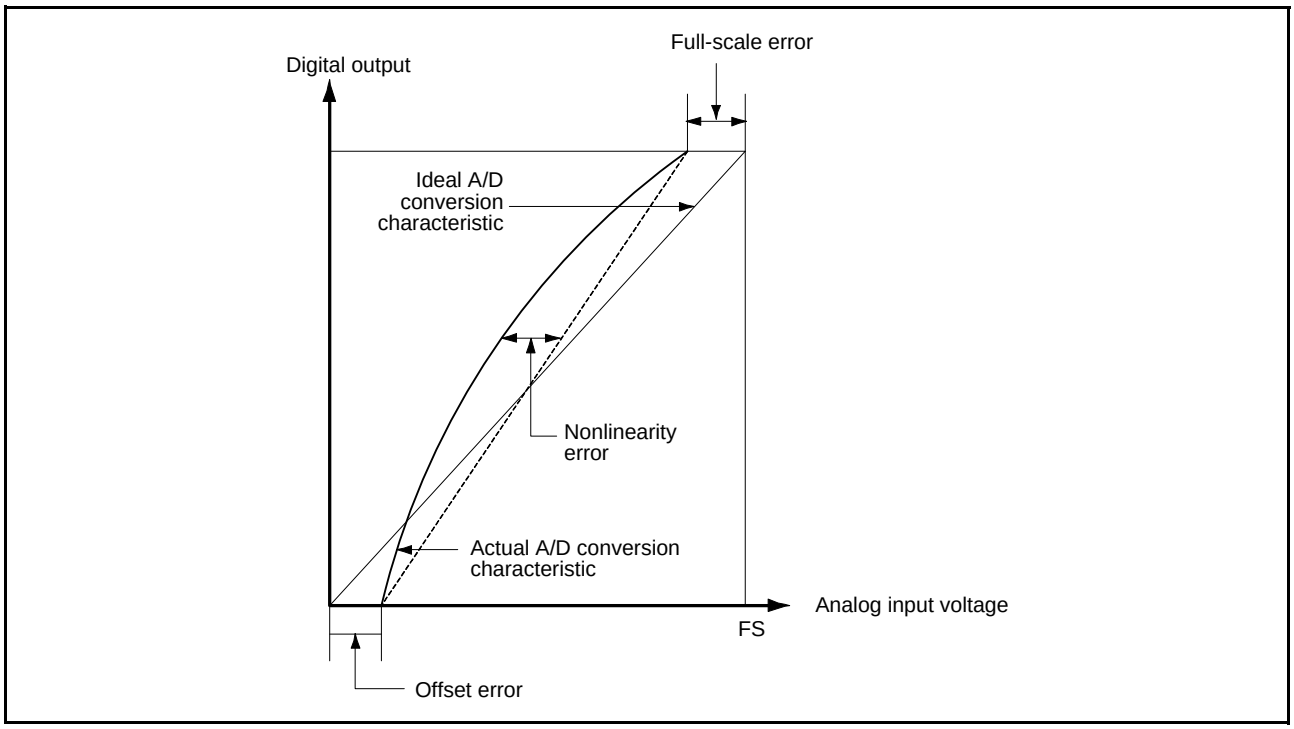


Figure 41.15 A/D Conversion Accuracy Definitions (2)

41.6 Usage Notes

41.6.1 Setting for the Module-Stop State

Module-stop control registers can used to enable or disable operation of the 10-bit A/D converter. The 10-bit A/D converter is stopped in the initial state. The registers become accessible on release from the module-stop state. For details, refer to section 11, Low Power Consumption.

41.6.2 Notes on Forcibly Stopping A/D Conversion

If an asynchronous trigger (ADTRG# pin) or a synchronous trigger (from the MTU, TPU, or TMR modules) is selected as the condition for the start of A/D conversion, unintended A/D conversion may still start even if the ADCSR.ADST bit has been set to 0 to forcibly stop A/D conversion.
Therefore, to make sure that the A/D converter is stopped, set the ADCR.TRGS[2:0] bits to 000b to select the software trigger as the condition for the start of A/D conversion and then clear the ADCSR.ADST bit to 0 (to stop A/D conversion).

41.6.3 A/D Conversion Restart Timing after Forced Stop

When A/D conversion is forcibly stopped by clearing the ADCSR.ADST bit to 0, one ADCLK cycle of time is required to stop the analog section of the 10-bit A/D converter.
When A/D conversion is started by setting the ADCSR.ADST bit to 1 while the analog section of the 10-bit A/D converter has been stopped, A/D conversion restarts after one ADCLK cycle has elapsed.

41.6.4 Notes on Entering Power-Down States

When the RX63N/RX631 Group device enters the module stop state or software standby mode with A/D conversion enabled, the analog power supply current value becomes the same as that used during A/D conversion. To reduce the analog power supply current in the module stop state or software standby mode, disable A/D conversion by clearing the ADCSR.ADST bit to 0. Note that to halt A/D conversion, sufficient time is required after the ADST bit is cleared to the stop of 10-bit A/D converter analog cycle. To ensure this, follow the procedure below.

1. Set the TRGS[2:0] bits in ADCR to 000b (software trigger).
2. Clear the ADST bit in ADCSR to 0.
3. Set the CKS[1:0] bits in ADCR to 11b (PCLK).
4. Ensure A/D conversion is stopped, then enter the device in the module stop state or software standby mode.

41.6.5 Notes on Performing 10-Bit A/D Converter Synchronous D/A Conversion

To perform 10-bit A/D converter synchronous D/A conversion to avoid DA to AD interference by setting the DAADST bit in the D/A A/D synchronous start control register (DAADSCR) of the D/A converter to 0, set the ADCR.TRGS[2:0] bits to 000b (software trigger), and then set the D/A control register (DACR) of the D/A converter while the ADCSR.ADST bit is 0.
During A/D conversion, the clock should not be halted due to module stop or other reasons. This may result in that D/A conversion as well as A/D conversion is halted. When A/D conversion is halted by setting the ADCSR.ADST bit to 0, one ADCLK cycle of time is required to halt the analog section of the 10-bit A/D converter.
When A/D conversion is restarted by setting the ADCSR.ADST bit to 1 immediately after setting the ADCSR.ADST bit to 0, A/D conversion is restarted after one ADCLK cycle period of time has elapsed.

41.6.6 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 1.0 kΩ or less to implement high conversion speed of 1.0 µs. If a capacitor with large capacitance is connected for only one pin conversion in single channel mode, the input load will essentially comprise only the internal input resistance Rs, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater) (see Figure 41.16). When converting a high-speed analog signal or converting multiple pins in scan mode, a low impedance buffer should be inserted.

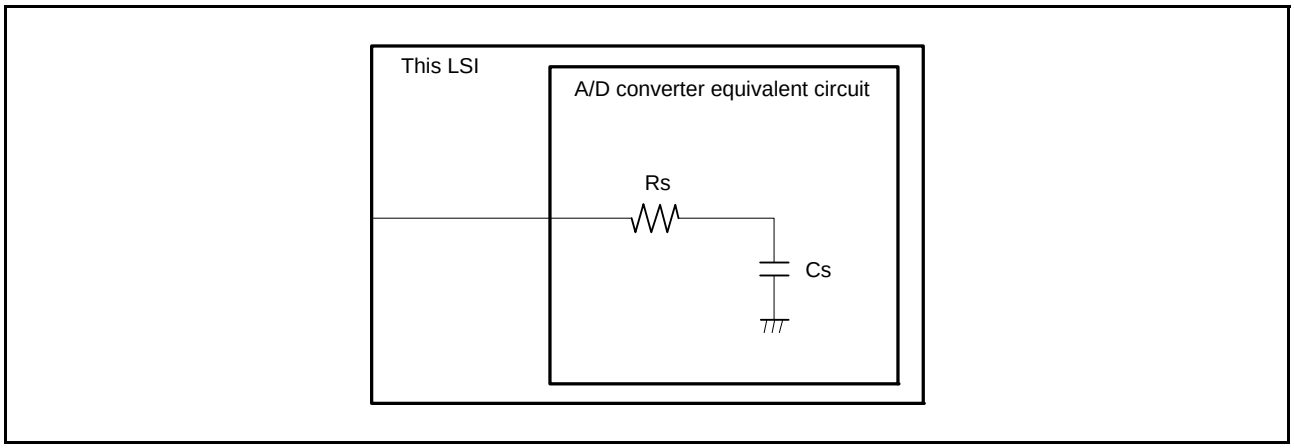


Figure 41.16 Internal Equivalent Circuit for Analog Input Pins

Table 41.9 Analog Pin Specifications

Item		Min.	Max.	Unit
Permissible signal source impedance		—	1.0	kΩ
Internal equivalent circuit for pins	Rs	—	6.5	kΩ
	Cs	—	6	pF

41.6.7 Influences on Absolute Precision

Adding capacitor results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND.
Care is also required to ensure that filter circuits do not interfere with digital signals or act as antennas on the mounting board.

41.6.8 Analog Power Supply Pin Setting Range

If this LSI is used outside the range of the following voltage settings, the reliability of the LSI may be adversely affected.

- Analog Input Voltage Range
  The voltage applied to analog input pin (AN0 to AN7) should be in the range VREFL ≤ VAN ≤ VREFH.
- Relationship between power supply pins (VREFH - VREFL, VCC - VSS)
  Establish the relationship between the reference voltage input pins (VREFH, VREFL) and the digital power supply pins (VCC, VSS) so that VREFH ≤ VCC and VREFL = VSS. As shown in Figure 41.17, connect a 0.1- F capacitor between each of the power supplies to form a closed loop in the shortest way so that VREFL = VSS in the source side. When the A/D converter is not used, set VREFH = VCC and VREFL = VSS.

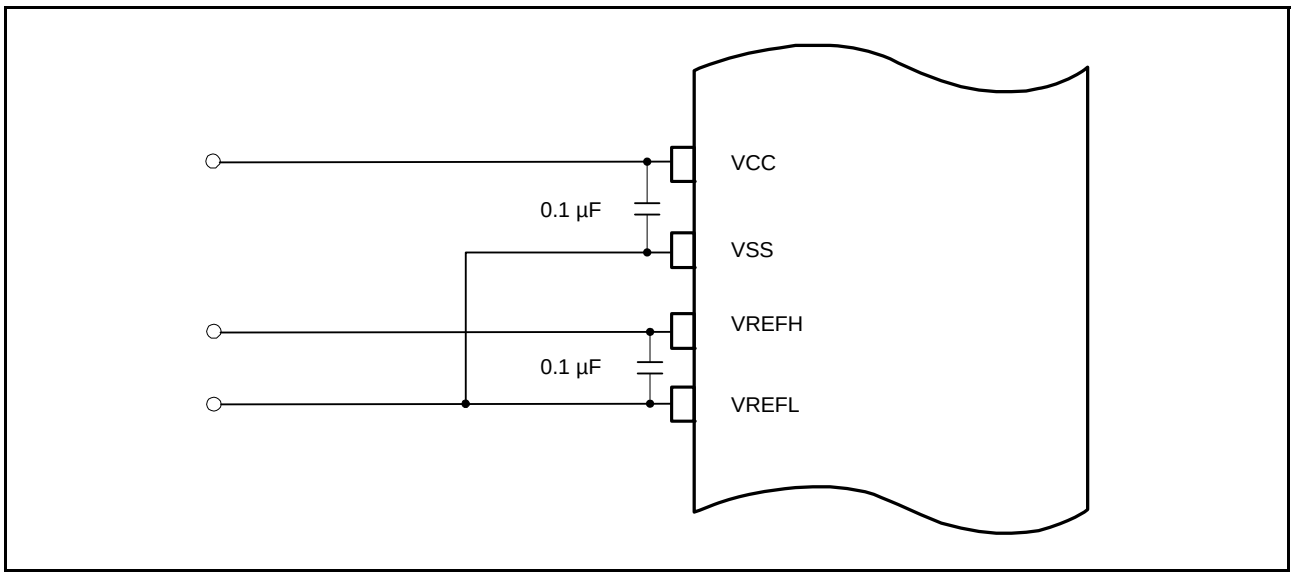


Figure 41.17 Connection Example of Power Supply Pins

41.6.9 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and the layout in which the digital circuit signal lines and analog circuit signal lines cross or are in close proximity to each other should be avoided as much as possible. Failure to do so may result in the incorrect operation of the analog circuitry due to inductance, adversely affecting the A/D conversion values.
In addition, digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog power supply (VREFH), and analog ground (VREFL). The analog ground (VREFL) should be connected at one point to a stable digital ground (VSS) on the board.

41.6.10 Notes on Noise Countermeasures

To prevent damage due to an abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN7), connect capacitors between the VREFH and VREFL, and a protection circuit relative to the analog input pins (AN0 to AN7), as shown in Figure 41.18.

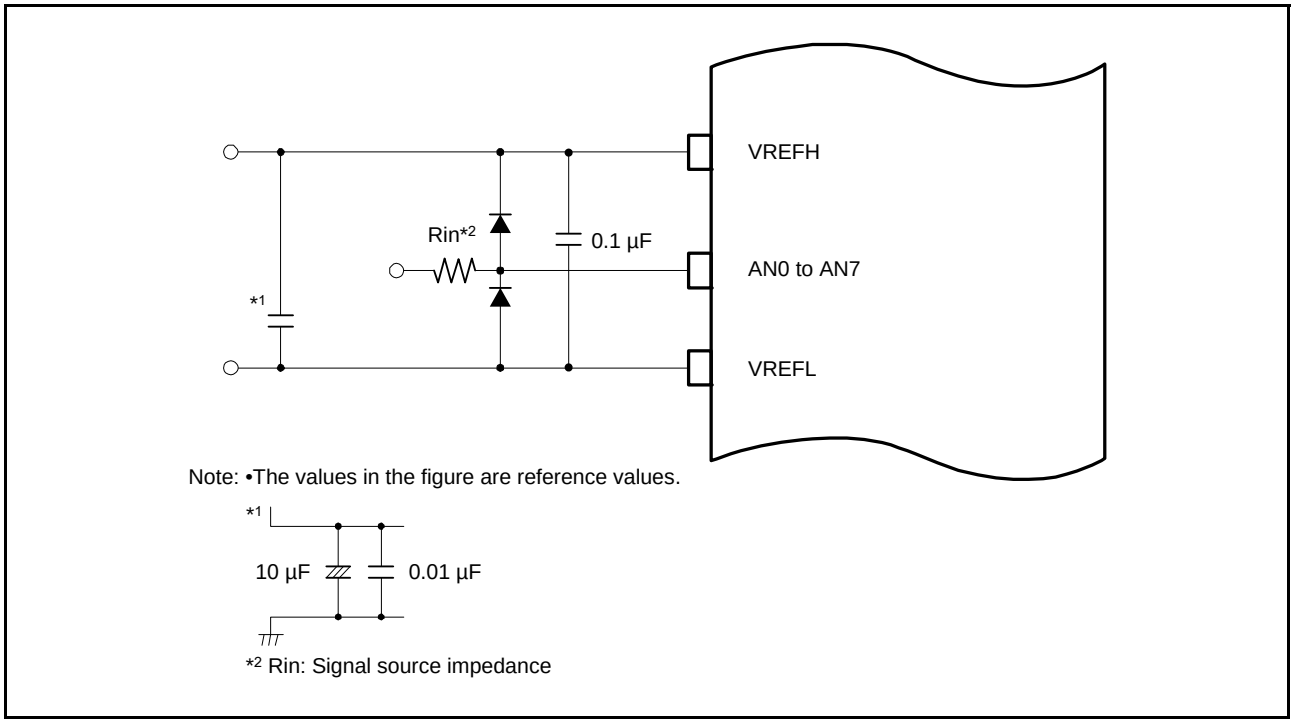
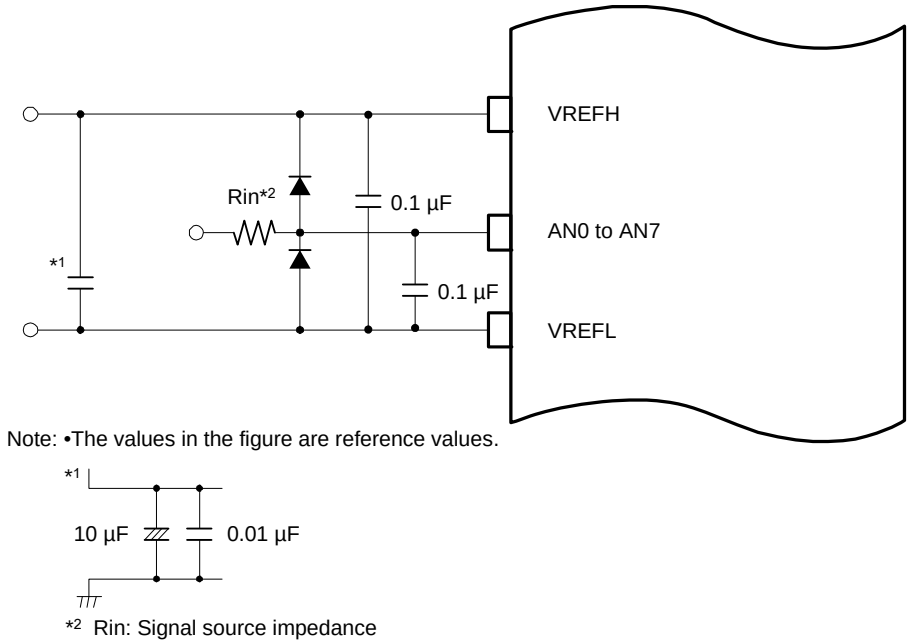


Figure 41.18 Example of Analog Input Pin Protection Circuit

41.6.11 Realizing High-Speed Conversion

To realize high-speed conversion, connect external 0.1-µF capacitors between the analog input pins (AN0 to AN7) and VREFL. This is shown in Figure 41.19. However, to hide the impedance of the signal source due to the input capacitance of the sample-and-hold circuit of the 10-bit A/D converter, the externally connected capacitors must be fully charged before the start of conversion.
Furthermore, when the voltages on the analog input pins fluctuate due to scanning and so on, so describing renewal of the charge of the externally connected capacitors, such full charge is beyond the scope of this description.



Note: •The values in the figure are reference values.

*2 Rin: Signal source impedance

Figure 41.19 Example of an Externally Connected Capacitor for High-Speed Conversion

42. D/A Converter (DAa)

42.1 Overview

The RX63N/RX631 Group includes two-channels of 10-bit D/A converter.
Table 42.1 lists the specifications of the D/A converter and Figure 42.1 shows a block diagram of the D/A converter.

Table 42.1 Specifications of D/A Converter

Item	Specifications
Resolution	10 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 10-bit A/D converter synchronous D/A conversion enable input signal from the 10-bit A/D converter (degradation of A/D conversion accuracy caused by interference is reduced by controlling the D/A converter inrush current generation timing with the enable signal).
Low-power consumption function	Module stop state can be set for each unit.

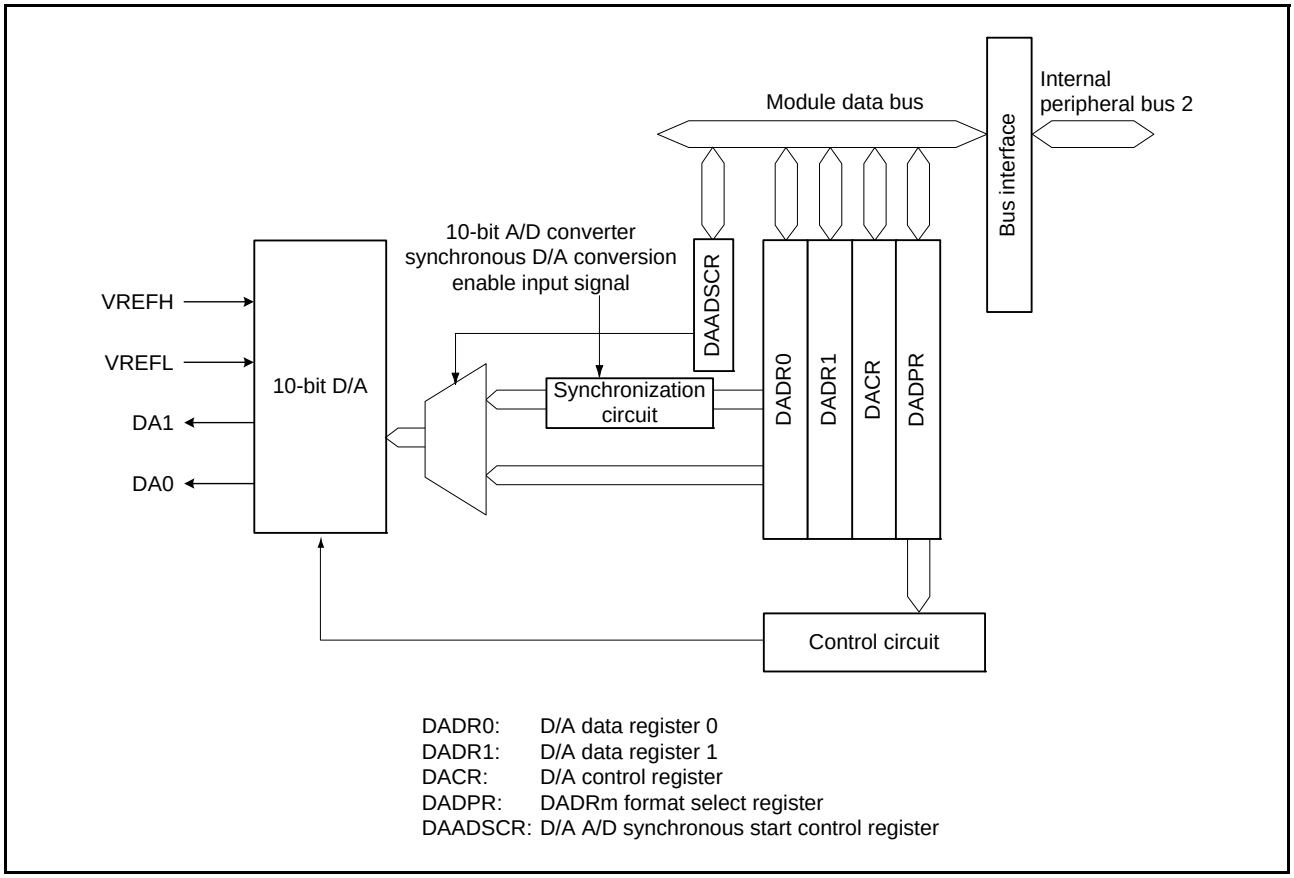


Figure 42.1 Block Diagram of D/A Converter

Table 42.2 lists the pin configuration of the D/A converter.

Table 42.2 Pin Configuration of D/A Converter

Pin Name	I/O	Function
VREFH	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This pin is also used as an analog power supply pin for each module. Connect to VCC when neither of these modules are used.
VREFL	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This pin is also used as an analog ground pin for each module. Set VREFL to the same potential as VSS.
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

42.2 Register Descriptions

42.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DADR0 0008 80C0h, DADR1 0008 80C2h

• DADPR.DPSEL bit = 0 (data are 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 are 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

DADRm registers are 16-bit readable/writable registers, which store 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.
10-bit data can be relocated by setting the DPSEL bit in DADPR.
Bits "—" are read as 0. The write value should be 0.

42.2.2 D/A Control Register (DACR)

Address(es): 0008 80C4h

	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 always be 1.	R/W
b5	DAE*1	D/A Enable	0: D/A conversion is independently controlled on channels 0 and 1. 1: D/A conversion on channels 0 and 1 is controlled as a single whole.	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.*2	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.*2	R/W

Note 1. This bit controls D/A conversion in combination with the DAOE0 and DAOE1 bits. The DAOE0 and DAOE1 bits control output of the results of conversion. For details, see Table 42.3.
Note 2. Set the P0.PDR.Bm bit (m = 3, 5) for pins used as analog outputs and the corresponding P0.PMR.Bm bit (m = 3, 5) to 0. In addition, set them to analog pins by the P03PFS and P05PFS registers. For details, see section 20, I/O Ports, and section 20, Multi-Function Pin Controller (MPC).

Table 42.3 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 and DA1) is enabled.
1	0	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is disabled.*1
		1	D/A conversion of channels 0 and 1 is enabled.sss Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.*1
	1	0	D/A conversion of channels 0 and 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 and DA1) is enabled.

Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

This register should be set while the 10-bit A/D converter is halted when the DAADSCR.DAADST bit is 1 (enabling the measure against interference between D/A and A/D conversion) (The register should be set while the ADCSR.ADST bit is 0 after the software trigger is selected as the trigger source for the 10-bit A/D converter).

DAE Bit (D/A Enable)

The DAE bit controls D/A conversion in combination with the DAOEi (i = 0 or 1) bit.

When the DAE bit is 0, D/A conversion is independently controlled on channels 0 and 1. When the DAE bit is 1, D/A conversion on channels 0 and 1 is controlled as a single whole. The DAOEi bit controls output of the results of conversion.

When the measure against interference for D/A and A/D conversion is enabled (the DAADSCR.DAADST bit = 1), set this bit while the ADCSR.ADST bit is set to 0. To ensure that the 10-bit A/D converter remains stopped while the setting is made, select the software trigger as the trigger source in advance.

DAOE0 Bit (D/A Output Enable 0)

The DAOE0 bit controls the D/A conversion and analog output.

When the measure against interference between D/A and A/D conversion is enabled (the DAADSCR.DAADST bit = 1), set this bit while the ADCSR.ADST bit is set to 0. To ensure that the 10-bit A/D converter remains stopped while the setting is made, select the software trigger as the trigger source in advance.

DAOE1 Bit (D/A Output Enable 1)

The DAOE1 bit controls the D/A conversion and analog output.

When the measure against interference between D/A and A/D conversion is enabled (the DAADSCR.DAADST bit = 1), set this bit while the ADCSR.ADST bit is set to 0. To ensure that the 10-bit A/D converter remains stopped while the setting is made, select the software trigger as the trigger source in advance.

42.2.3 DADRm Format Select Register (DADPR)

Address(es): 0008 80C5h

	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 always read as 0. The write value should always 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

42.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): 0008 80C6h

	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: D/A converter operation does not synchronize with 10-bit A/D converter operation. (measure against interference between D/A and A/D conversion is disabled) 1: D/A converter operation synchronizes with 10-bit A/D converter 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, DAADSCR switches synchronization of the timing of the start of D/A conversion with the 10-bit A/D converter synchronous D/A conversion enable input signal off or on. This register should be set while the 10-bit A/D converter is halted (while the ADCSR.ADST bit is 0 after selecting software trigger as the 10-bit A/D converter trigger).

DAADST Bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRm register value to be converted into analog data at any time. Setting the DAADST bit to 1 allows synchronous D/A conversion with the synchronous D/A conversion enable signal from the A/D converter. Therefore, even if the DADRm register value is modified, D/A conversion does not start until the A/D converter completes A/D conversion.

Set this bit while the ADCSR.ADST bit is set to 0. To ensure that the 10-bit A/D converter remains stopped while the setting is made, select the software trigger as the trigger source in advance.

42.3 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOEi bit (i = 0, 1) in DACR 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 42.2 shows the timing of this operation.

1. Write the data for conversion to DADR0.
2. Set the DAOE0 bit in DACR 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 DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{Setting value of DADR0}}{1024} \times \text{VREFH}$$

3. If DADR0 is written to again, the conversion is immediately 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 cleared to 0, analog output is disabled.

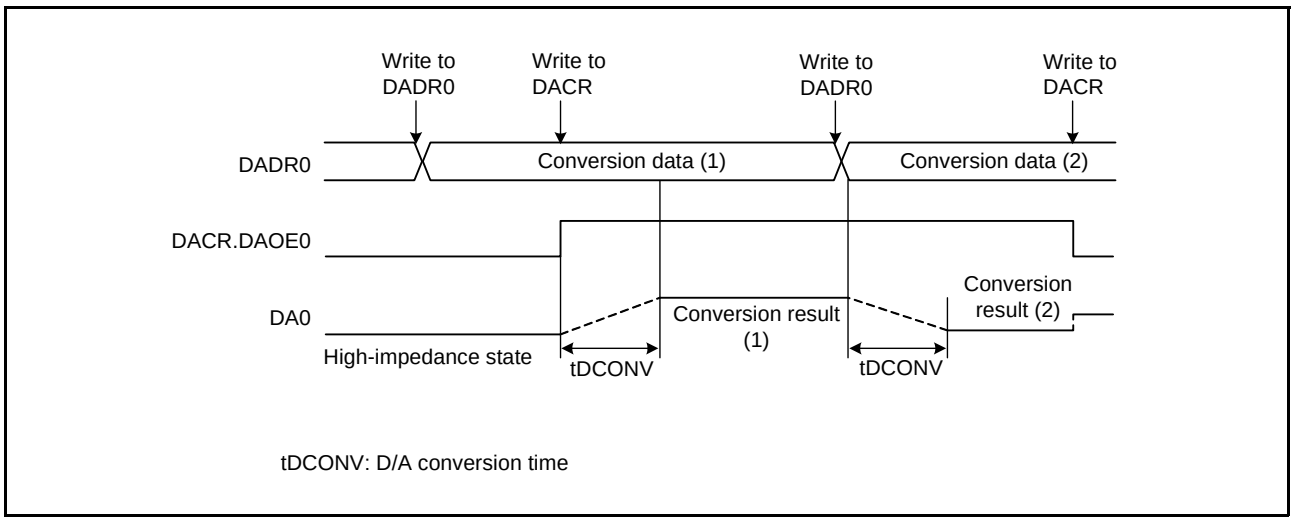


Figure 42.2 Example of D/A Converter Operation

42.3.1 Measure against Interference between D/A and A/D Conversion

When D/A conversion starts, the D/A converter generates inrush current. Since the same analog power supply is shared by the D/A converter and 10-bit A/D converter, the generated inrush current may interfere with 10-bit A/D converter operation.

To prevent such interference, the D/A converter start timing can be synchronized with the 10-bit A/D converter synchronous D/A conversion enable signal.

With the DAADSCR.DAADST bit being 1, even if the DADRm register data is modified during 10-bit A/D converter 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 DAADSCR.DAADST being 1, when the DADRm register data is modified while the 10-bit A/D converter is halted, D/A conversion starts in one PCLK cycle.

The following describes an example of channel 0 D/A conversion, in which the D/A converter operates synchronously with the 10-bit A/D converter.

(1) Confirm that the 10-bit A/D converter is halted. Set the DAADSCR.DAADST bit to 1.
(2) Confirm that the 10-bit A/D converter is halted. Set the DACR.DAOE0 bit to 1.
(3) Set the DADR0 register.

- If the 10-bit A/D conversion is halted when the DADR0 register is modified, D/A conversion starts in one PCLK cycle.
- If the 10-bit A/D conversion is in progress 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.

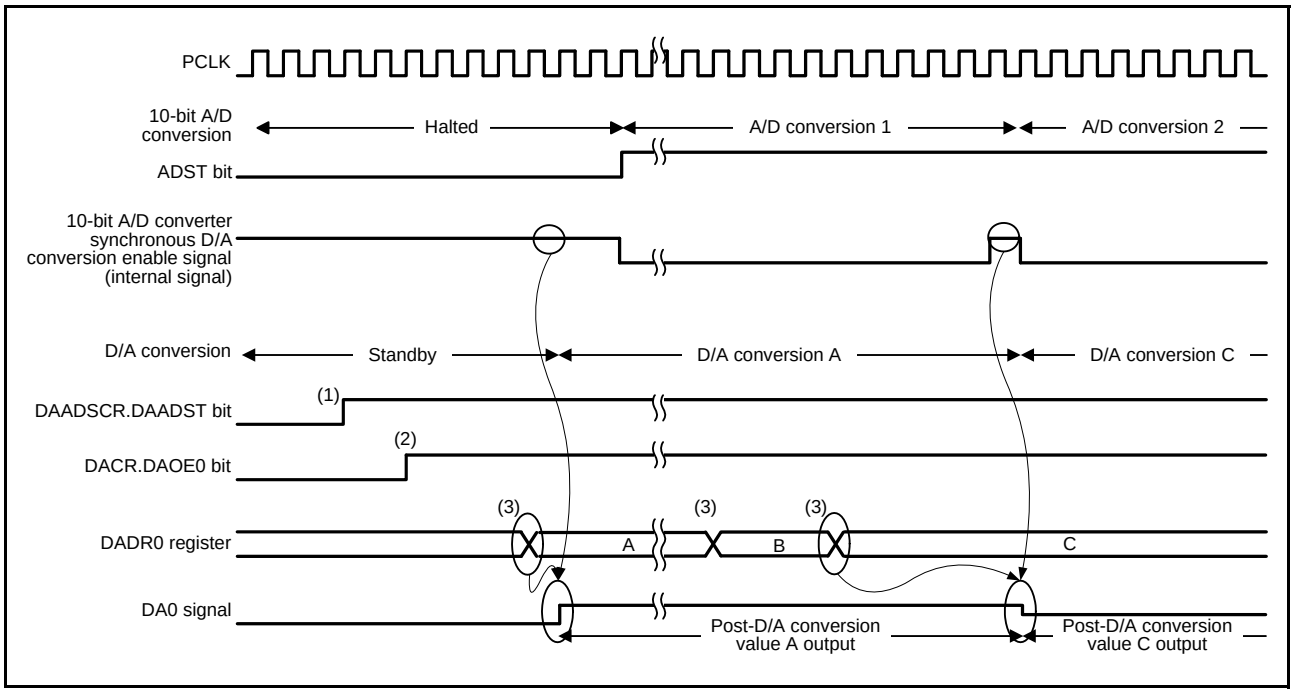


Figure 42.3 Example of Conversion when the D/A Converter is Synchronized with the 10-Bit A/D Converter

42.4 Usage Notes

42.4.1 Module Stop Function Setting

Operation of the D/A converter can be disabled or enabled by using the module stop control register. The initial setting is for operation of the D/A converter to be halted. Register access is enabled by clearing the module stop state. For details, refer to section 11, Low Power Consumption.

42.4.2 Operation of the D/A Converter in Module Stop State

When the RX63N/RX631 Group enters the module stop state with D/A conversion enabled, the D/A 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 clearing the DAOE1, DAOE0, and DAE bits in DACR to 0.

42.4.3 Operation of the D/A Converter in Software Standby Mode

When the RX63N/RX631 Group enters software standby mode with D/A conversion enabled, the D/A 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 clearing the DAOE1, DAOE0, and DAE bits in DACR to 0.

42.4.4 Note on Entering Deep Software Standby Mode

When the RX63N/RX631 Group enters deep software standby mode with D/A conversion enabled, the outputs of the D/A converter are placed in a high impedance state.

42.4.5 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 10-bit A/D converter into the module stop state. It may halt D/A conversion in addition to A/D conversion.

42.4.6 Notes on Usage when Measure against Interference between D/A and A/D Conversion is Disabled

The 10-bit A/D converter synchronous D/A conversion enable input signal is used when the 10-bit A/D converter and the D/A converter share the same analog power-supply pin. If this is not the case, fix this input signal to 1 because it is not in use, and write 0 to the D/A A/D synchronous conversion bit.

43. Temperature Sensor

43.1 Overview

The RX63N/RX631 Group includes a temperature sensor. The temperature sensor outputs a voltage which varies with the temperature. The 12-bit A/D converter can convert the voltage from the sensor into a digital value. The user can then obtain the temperature around the LSI by converting the value into the temperature.
Table 43.1 lists the specifications of the temperature sensor, and Figure 43.1 shows a block diagram of the temperature sensor.

Table 43.1 Specifications of Temperature Sensor

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter.
Low-power consumption function	The module stop state is selectable.

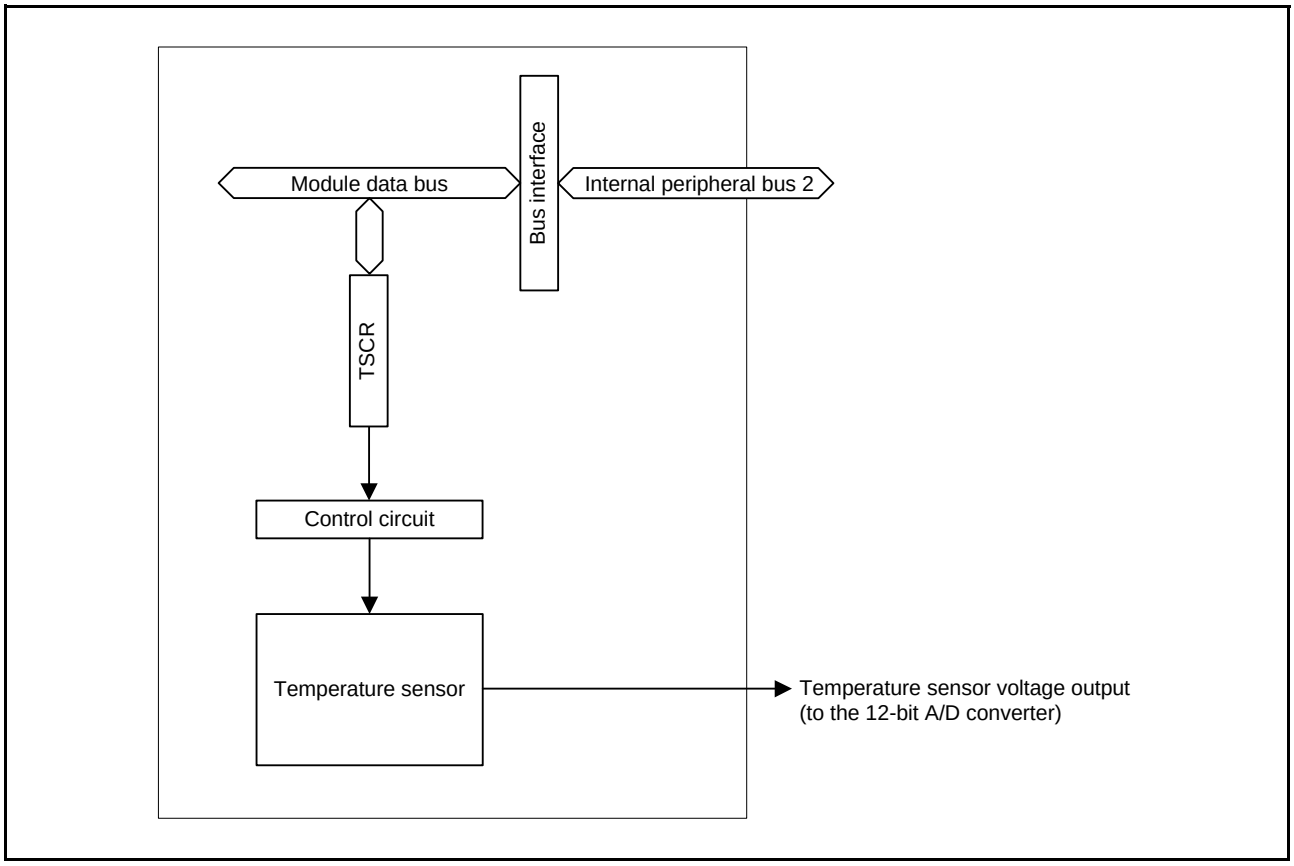


Figure 43.1 Block Diagram of Temperature Sensor

43.2 Register Descriptions

43.2.1 Temperature Sensor Control Register (TSCR)

Address: 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. 1: Enables output from the temperature sensor to the 12-bit A/D converter.	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

43.3 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. The user can then obtain the temperature around the LSI by converting the value into the temperature.

43.3.1 Preparation for Using the Temperature Sensor

The temperature characteristics of the temperature sensor are 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 at the time of temperature measurement (V)
T1: Temperature experimentally measured at one point (°C)
V1: Voltage output by the temperature sensor at the time of measurement of T1 (V)
T2: Temperature at the experimental measurement of another point (°C)
V2: Voltage output by the temperature sensor at the time of measurement of T2 (V)
Slope: Temperature gradient by the temperature sensor (V/°C); slope = (V2 – V1)/(T2 – T1)

Characteristics vary from sensor to sensor. Therefore, the following experimental measurement at two different temperatures is recommended.
Use the 12-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
Again, using the 12-bit A/D converter, measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature gradient (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 gradient given in Table 45.XX of section 49, Electrical Characteristics, use the 12-bit A/D converter 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 method gives less accurate temperatures than measurement at two points.

T = (Vs – V1)/slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor at the time of temperature measurement (V)
T1: Temperature experimentally measured at one point (°C)
V1: Voltage output by the temperature sensor at the time of measurement of T1 (V)
Slope: Temperature gradient (V/°C) given in Table 45.25.

43.3.2 Setting of 12-Bit A/D Converter

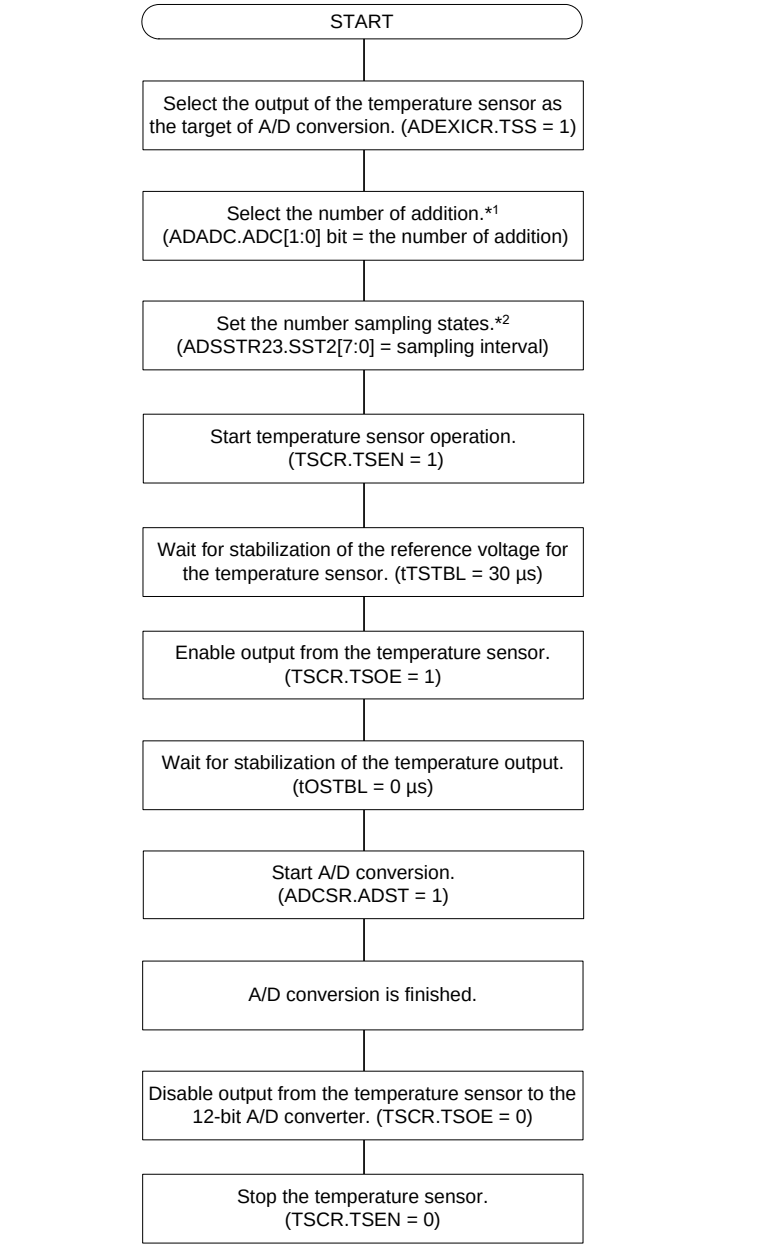
For A/D conversion of temperature sensor output voltages, 12-bit A/D converter registers should be set as follows.

- 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.TSS) to 1. Additionally, remove other sources from the scope of conversion by setting all of the bits in the A/D channel select registers 0 and 1 (ADANS0 and ADANS1), and the internal reference voltage A/D select bit in the A/D conversion extended input control register (ADEXICR.OCS), to 0.
- Setting Single Scan Mode
  Select single scan mode by setting the scan mode select bit in the A/D control register (ADCSR.ADCS) to 0.
- Setting Addition Mode
  For A/D conversion of the temperature sensor output, addition mode is selectable. To enable addition 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 for addition.
- Setting the Number of Sampling States of the 12-Bit A/D Converter
  The number of “states” for sampling of the output of the temperature sensor for A/D conversion is selectable. The initial setting is for 20 states. To change the number of states for sampling from 20 states, set the sampling time 2 setting bits in the A/D sampling state register 23 (ADSSTR23.SST2[7:0]).

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 43.3.3, Procedure for Using the Temperature Sensor.

43.3.3 Procedure for Using the Temperature Sensor

Figure 43.2 shows the procedure for using the temperature sensor.



Note 1. Unless addition mode is set, this setting is not required.
Note 2. The initial value for the number of states over which sampling proceeds is 20.
If the number of states is not to be changed, this setting is not required.

Figure 43.2 Procedure for Using the Temperature Sensor

43.3.4 Timing of A/D Conversion of Temperature Sensor Output

Figure 43.3 is a chart of the timing from the start of temperature-sensor operation until the completion of A/D conversion. Times in the chart are described in Table 43.2.

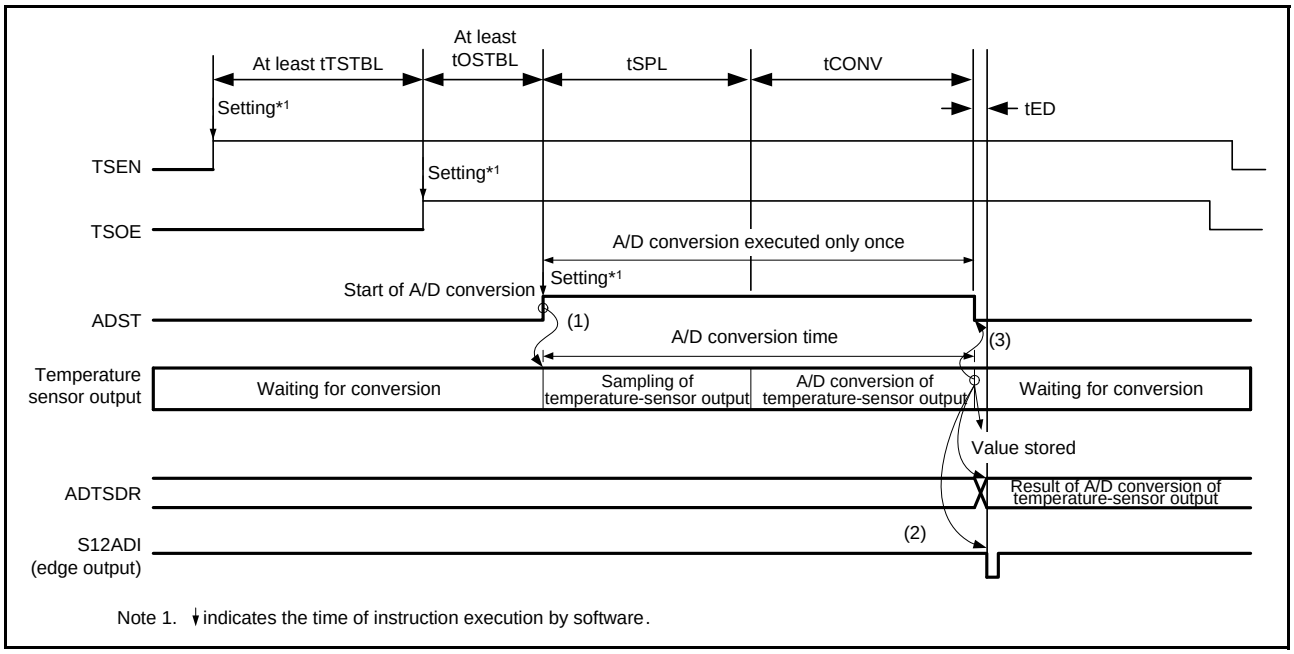


Figure 43.3 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion

Table 43.2 Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation

Item	Symbol	Time
Waiting time for temperature-sensor reference-voltage stabilization	tTSTBL	30 µs (max)
Waiting time for temperature-sensor output stabilization	tOSTBL	0 µs (max)
12-bit A/D converter input sampling time	tSPL	ADSSTR23 setting x PCLK cycles
A/D conversion time	tCONV	Refer to Table37.8, Time Required for scanning Operations in section 37.3.4, Analog Input Sampling and Scan Conversion Time.l
Scan conversion end delay time	tED	Refer to Table37.8, Time Required for scanning Operations in section 37.3.4, Analog Input Sampling and Scan Conversion Time.l

43.4 Usage Note

43.4.1 Module Stop Function Setting

The corresponding bit in the 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.

44. RAM

The RX63N Group, RX631 Group has an on-chip high-speed static RAM.

44.1 Overview

Table 44.1 lists the specifications of the RAM.

Table 44.1 Specifications of RAM

Item	Description
RAM capacity	128 Kbytes (RAM0: 64 Kbytes, RAM1: 64 Kbytes)
RAM address	RAM0: 0000 0000h to 0000 FFFFh (64 Kbytes) RAM1: 0001 0000h to 0001 FFFFh (64 Kbytes)
Access	• Single-cycle access is possible for both reading and writing. • On-chip RAM can be enabled or disabled.*1
Data retention function	Data in RAM0 can be retained in deep standby mode.
Low-power consumption function	The module stop state is independently selectable for RAM0 and RAM1.

Note 1. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.4, System Control Register 1 (SYSCR1).

44.2 Operation

44.2.1 Data Retention

The address space for on-chip RAM is divided into the RAM0 and RAM1 areas. The difference between the two is whether internal power can be supplied in deep software standby mode.
Whether or not the supply of internal power to RAM0 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 RAM0 are retained in deep software standby mode. The supply of internal power supply to RAM1 is stopped at this time, so data are not retained in RAM1.
See section 11, Low Power Consumption, for details on the DPSBYCR.DEEPCUT[1:0] bits.

44.2.2 Low-Power Consumption Function

Power consumption can be reduced by setting the module stop control register C (MSTPCRC) to stop supply of the clock signal to the on-chip RAM.
If the MSTPC0 bit in MSTPCRC is set to 1, supply of the clock signal to RAM0 is stopped. If the MSTPC1 bit in MSTPCRC is set to 1, supply of the clock signal to RAM1 is stopped.
The respective modules (RAM0 and RAM1) are 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.
For details on the MSTPCRC registers, see section 11, Low Power Consumption.

45. ROM (Flash Memory for Code Storage)

The RX63N/RX631 Group has a maximum 2-Mbyte flash memory for storing code (ROM).
This section explains the flash memory for code storage. For details on the E2 DataFlash, see section 46, E2 DataFlash Memory (Flash Memory for Data Storage).

45.1 Overview

Table 45.1 lists the specifications of the ROM, Table 45.2 lists the correspondence between ROM capacity and ROM addresses, and Figure 45.1 shows a block diagram of the ROM, E2 DataFlash, and related modules.

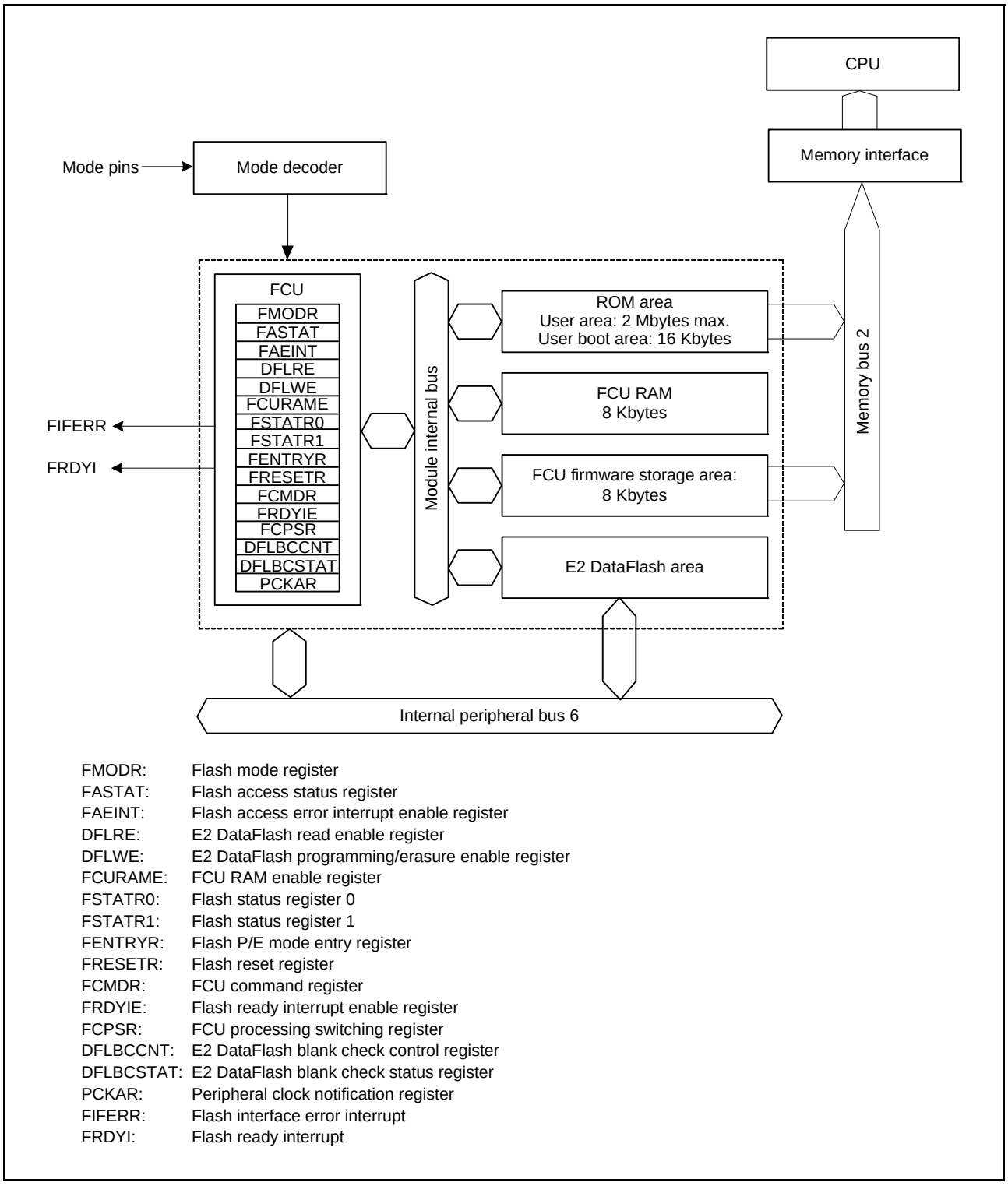
Table 45.1 Specifications of ROM

Item		Specifications
Memory space		User area: 2 Mbytes max. User boot area: 16 Kbytes
High-speed reading		A read operation takes one cycle of ICLK
Programming/erasing method		• The chip incorporates a dedicated sequencer (FCU) for programming of the ROM. • Programming and erasing the ROM are handled by issuing commands to the FCU. • FFFF FFFFh is read from the erased ROM in 32 bits.
BGO (background operation)		• Execution of program code from the ROM is possible while the E2 DataFlash memory is being programmed or erased. • The CPU is able to execute program code from areas other than the ROM or E2 DataFlash while the ROM is being programmed or erased.
Suspension and resumption		• The CPU is able to execute program code from the ROM during suspension of programming or erasure. • Programming and erasure of the ROM can be restarted (resumed) after suspension.
Units of programming and erasure		• Units of programming for the user area or user boot area: 128 bytes • Units of erasure for the user area: In block units • Units of erasure for the user boot area: 16 Kbytes
On-board programming (four types)	Boot mode	• The user area or user boot area is programmable via the SCI. • The bit rate for SCI transfer between the host and RX63N/RX631 is automatically adjusted.
	USB boot mode	The user area is programmable via the USB.
	User boot mode	The user area can be programmed after the system is started using the user boot area.
	User program mode	The user program can be used to program the user area.
Off-board programming		A PROM programmer can be used to program the user area and user boot area.
Protection	Software-controlled protection	The FENTRYR.FENTRY3, FENTRY2, FENTRY1, FENTRY0*1, FWEPROR.FLWE[1:0], and lock bits can be set to prevent unintentional programming.
	Error protection	When abnormal operations are detected during programming/erasure, this function disables any further programming/erasure.
Programming/erasure time, number of programming		See section 49, Electrical Characteristics.

Note 1. The FENTRY1, FENTRY2, and FENTRY3 bits are provided when the user area capacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively.

Table 45.2 Correspondence between ROM Capacity and ROM Addresses

ROM Capacity	ROM Addresses
768 Kbytes	FFF4 0000h to FFFF FFFFh
1 Mbyte	FFF0 0000h to FFFF FFFFh
1.5 Mbytes	FFE8 0000h to FFFF FFFFh
2 Mbytes	FFE0 0000h to FFFF FFFFh



FMODR: Flash mode register
FASTAT: Flash access status register
FAEINT: Flash access error interrupt enable register
DFLRE: E2 DataFlash read enable register
DFLWE: E2 DataFlash programming/erasure enable register
FCURAME: FCU RAM enable register
FSTATR0: Flash status register 0
FSTATR1: Flash status register 1
FENTRYR: Flash P/E mode entry register
FRESETR: Flash reset register
FCMDR: FCU command register
FRDYIE: Flash ready interrupt enable register
FCPSR: FCU processing switching register
DFLBCCNT: E2 DataFlash blank check control register
DFLBCSTAT: E2 DataFlash blank check status register
PCKAR: Peripheral clock notification register
FIFERR: Flash interface error interrupt
FRDYI: Flash ready interrupt

Figure 45.1 Block Diagram of ROM

Input and output pins associated with the ROM are listed in Table 45.3.

Table 45.3 Input and Output Pins Associated with the ROM

Pin Name	I/O	Description
PF2/RXD1 (177/176-pin packages) P30/RXD1 (145/144/100-pin packages)	Input	Used in boot mode to receive data through SCI (for host communications)
PF0/TXD1 (177/176-pin package) P26/TXD1 (145/144/100-pin packages)	Output	Used in boot mode to transmit data through SCI (for host communications)
MD	Input	Selection of operating mode
PC7	Input	Selection of boot mode (SCI boot), USB boot mode or user boot mode
USB0_DP, USB0_DM	I/O	Data input/output of USB (for use in USB boot mode)
P14/USB0_DPUPE	I/O	Pull-up pin for USB (for use in USB boot mode)
P16/USB0_VBUS	Input	Detection of connection and disconnection of USB cables (for use in USB boot mode)
P35	Input	Selection of USB bus-power mode or self-power mode (for use in USB boot mode)

45.2 Register Descriptions

Although some registers include bits related to E2 DataFlash, this section deals only with the bits related to ROM. For details on the bits related to the E2 DataFlash, see section 46.2, Register Descriptions (in section 46, E2 DataFlash Memory (Flash Memory for Data Storage)).
P/E indicates programming and erasure.

45.2.1 Flash Write Erase Protection 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 to b0	FLWE[1:0]	Flash Programming/ Erasure	b1 b0 0 0: Flash programming/erasure disabled. 0 1: Flash programming/erasure enabled. 1 0: Flash programming/erasure disabled. 1 1: Flash programming/erasure disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FWEPROR is initialized in software standby mode or deep software standby.

FLWE[1:0] Bits (Flash Programming/Erasure)

These bits protect the execution of the flash programming/erasure with software.

45.2.2 Flash Mode Register (FMODR)

Address(es): 007F C402h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FRDMD	—	—	—	—
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	FRDMD	FCU Read Mode Select	0: Memory Area Read Method Set this bit to 0 when reading ROM lock bits in ROM lock bit read mode. 1: Register Read Method Set this bit to 1 when using lock bit read 2 command to read ROM lock bits.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When the on-chip ROM is disabled, the data read from FMODR is 00h and writing is disabled.

FRDMD Bit (FCU Read Mode Select)

This bit is used to select a method for reading lock bits.

If the blank check command for the E2 DataFlash is to be used, this bit has to be set for the register read mode (see section 46, E2 DataFlash Memory (Flash Memory for Data Storage)).

45.2.3 Flash Access Status Register (FASTAT)

Address(es): 007F C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAE	—	—	CMDLK	DFLAE	—	DFLRPE	DFLWPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPE	E2 DataFlash Programming/Erasure Protection Violation	See section 46, E2 DataFlash Memory (Flash Memory for Data Storage).	R/(W) *1
b1	DFLRPE	E2 DataFlash Read Protection Violation	See section 46, E2 DataFlash Memory (Flash Memory for Data Storage).	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DFLAE	E2 DataFlash Access Violation	See section 46, E2 DataFlash Memory (Flash Memory for Data Storage).	R/(W) *1
b4	CMDLK	FCU Command Lock	0: FCU is not in the command-locked state 1: FCU is in the command-locked state	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAE	ROM Access Violation	0: No ROM access error 1: ROM access error	R/(W) *1

Note 1. Only 0 can be written after reading 1 to clear the flag to 0.

When on-chip ROM is disabled, the data read from FASTAT is 00h and writing is disabled. When one of the bits in FASTAT is set to 1, the FCU is placed in the command-locked state (see section 45.8.2, Error Protection). To clear the command-locked state, a status register clear command must be issued to the FCU after setting FASTAT to 10h.

CMDLK Bit (FCU Command Lock)

This bit indicates that the FCU is in the command-locked state (see section 45.8.2, Error Protection).

[Setting condition]

- After the FCU detects an error and enters the command-locked state

[Clearing condition]

- After the FCU issues a status register clear command under conditions where FASTAT is set to 10h

ROMAE Bit (ROM Access Violation)

This bit indicates whether a ROM access violation occurred.
When the ROMAE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, placing the FCU in the command-locked state.

[Setting conditions]

- Read access to a ROM programming/erasure address when the FCU is in ROM P/E normal mode*1

ROM Capacity	ROM programming/erasure address ranges			
	The FENTRY0 bit is 1	The FENTRY1 bit is 1	The FENTRY2 bit is 1	The FENTRY3 bit is 1
256 Kbytes	00FC 0000h to 00FF FFFFh	—	—	—
512 Kbytes	00F8 0000h to 00FF FFFFh	—	—	—
1 Mbyte	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	—	—
1.5 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	—
2 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	00E0 0000h to 00E7 FFFFh

- Access to a ROM programming/erasure address*1

ROM Capacity	ROM programming/erasure address ranges			
	The FENTRY0 bit is 0	The FENTRY1 bit is 0	The FENTRY2 bit is 0	The FENTRY3 bit is 0
256 Kbytes	00FC 0000h to 00FF FFFFh	—	—	—
512 Kbytes	00F8 0000h to 00FF FFFFh	—	—	—
1 Mbyte	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	—	—
1.5 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	—
2 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	00E0 0000h to 00E7 FFFFh

- Read access to a ROM-reading address while FENTRYR has placed the ROM in ROM P/E mode

ROM Capacity	ROM programming/erasure address ranges
256 Kbytes	FFFC 0000h to FFFF FFFFh
512 Kbytes	FFF8 0000h to FFFF FFFFh
1 Mbyte	FFF0 0000h to FFFF FFFFh
1.5 Mbytes	FFE8 0000h to FFFF FFFFh
2 Mbytes	FFE0 0000h to FFFF FFFFh

[Clearing condition]

- When 0 is written after reading 1

Note 1. The FENTRY1, FENTRY2, and FENTRY3 bits are provided when the user area capacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively.

45.2.4 Flash Access Error Interrupt Enable Register (FAEINT)

Address(es): 007F C411h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
Value after reset:	1	0	0	1	1	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPEIE	E2 DataFlash Programming/Erasure Protection Violation Interrupt Enable	See section 46, E2 DataFlash Memory (Flash Memory for Data Storage).	R/W
b1	DFLRPEIE	E2 DataFlash Read Protection Violation Interrupt Enable	See section 46, E2 DataFlash Memory (Flash Memory for Data Storage).	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DFLAEIE	E2 DataFlash Access Violation Interrupt Enable	See section 46, E2 DataFlash Memory (Flash Memory for Data Storage).	R/W
b4	CMDLKIE	FCU Command Lock Interrupt Enable	0: FIFERR interrupt requests disabled when the FASTAT.CMDLK bit is set to 1 1: FIFERR interrupt requests enabled when the FASTAT.CMDLK bit is set to 1	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAEIE	ROM Access Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the FASTAT.ROMAE bit is set to 1 1: FIFERR interrupt requests enabled when the FASTAT.ROMAE bit is set to 1	R/W

When on-chip ROM is disabled, the data read from FAEINT is 00h and writing is disabled.

CMDLKIE Bit (FCU Command Lock Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an FCU command lock occurs and the FASTAT.CMDLK bit is set to 1.

ROMAEIE Bit (ROM Access Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when a ROM access violation occurs and the FASTAT.ROMAE bit is set to 1.

45.2.5 FCU RAM Enable Register (FCURAME)

Address(es): 007F C454h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	FCRME
Value after 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	FCRME	FCU RAM Enable	0: Access to the FCU RAM disabled 1: Access to the FCU RAM enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	This bit is used to enable or disable rewriting of the FCRME bit.	R/(W) *1

Note 1. Write data is not retained.

Write access to FCURAME is valid only when the specific value is written to the upper byte in word access. Data written to the upper byte is not retained.
When on-chip ROM is disabled, the data read from FCURAME is 00h and writing is disabled.

FCRME Bit (FCU RAM Enable)

This bit is used to enable and disable an access to the FCU RAM.
Data written to the FCRME bit is valid only when it is written in word access and the KEY[7:0] bits are C4h. When writing to the FCU RAM, set FENTRYR to 0000h and stop the FCU.
The FCU RAM cannot be read regardless of whether access to it is enabled or disabled. Any value read from FCU RAM is always undefined.

45.2.6 Flash Status Register 0 (FSTATR0)

Address(es): 007F FFB0h

	b7	b6	b5	b4	b3	b2	b1	b0
	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRGSPD	Programming Suspend Status	0: Other than the status described below 1: During programming suspend processing or programming suspended	R
b1	ERSSPD	Erasure Suspend Status	0: Other than the status described below 1: When erasure suspend processing or erasure suspended	R
b2	—	Reserved	This bit is read as 0 and cannot be modified.	R
b3	SUSRDY	Suspend Ready	0: P/E suspend commands cannot be received 1: P/E suspend commands can be received	R
b4	PRGERR	Programming Error	0: Programming terminates normally 1: An error occurs during programming	R
b5	ERSERR	Erasure Error	0: Erasure terminates normally 1: An error occurs during erasure	R
b6	ILGLERR	Illegal Command Error	0: FCU detects no illegal command or illegal ROM/E2 DataFlash access 1: FCU detects an illegal command or illegal ROM/E2 DataFlash access	R
b7	FRDY	Flash Ready	0: During programming/erasure, During suspending programming/erasure, During the lock bit read 2 command processing, During the peripheral clock notification command processing, During the blank check processing of E2 DataFlash (See section 46, E2 DataFlash Memory (Flash Memory for Data Storage)). 1: Processing described above is not performed	R

When on-chip ROM is disabled, the data read from FSTATR0 is 00h.
FSTATR0 is initialized by a reset, or when the FRESETR.FRESET bit is set to 1

PRGSPD Bit (Programming Suspend Status)

This bit is used to indicate that the FCU enters the programming suspend processing state or programming suspended state. For details, see section 45.7, Suspending Operation.

[Setting condition]

- The FCU has initiated a write suspend command.

[Clearing condition]

- The FCU has accepted a resume command.

ERSSPD Bit (Erasure Suspend Status)

This bit is used to indicate that the FCU enters the erasure suspend processing state or erasure suspended state. For details, see section 45.7, Suspending Operation.

[Setting condition]

- The FCU has initiated an erasure suspend command.

[Clearing condition]

- The FCU has accepted a resume command.

SUSRDY Bit (Suspend Ready)

This bit is used to indicate whether the FCU can receive a P/E suspend command.

[Setting condition]

- After starting programming/erasure process, the FCU enters a state in which P/E suspend commands can be received.

[Clearing conditions]

- The FCU has accepted a P/E suspend command.
- During programming/erasure process, the FCU enters the command-locked state.

PRGERR Bit (Programming Error)

This bit is used to indicate the result of the ROM/E2 DataFlash programming process by the FCU.
When the PRGERR bit is set to 1, the FCU is placed in the command-locked state. For details, see section 45.8.2, Error Protection.

[Setting conditions]

- An error occurs during programming.
- A programming command is issued to areas protected by a lock bit.

[Clearing condition]

- After the FCU processes a status register clear command

ERSERR Bit (Erasure Error)

This bit is used to indicate the result of the ROM/E2 DataFlash erasure process by the FCU.
When the ERSERR bit is set to 1, the FCU is placed in the command-locked state. For details, see section 45.8.2, Error Protection.

[Setting conditions]

- An error occurs during erasure.
- A block erase command is issued to areas protected by a lock bit.

[Clearing condition]

- After the FCU processes a status register clear command

ILGLERR Bit (Illegal Command Error)

This bit is used to indicate that the FCU detects any illegal command or ROM/E2 DataFlash access.
When the ILGLERR bit is set to 1, the FCU is placed in the command-locked state. For details, see section 45.8.2, Error Protection.

[Setting conditions]

- The FCU detects an illegal command.
- The FCU detects an illegal ROM/E2 DataFlash access
  (one of the ROMAE, DFLAE, DFLRPE, and DFLWPE bits in FASTAT is 1).
- The setting of FENTRYR is invalid.

[Clearing condition]

- After the FCU processes a status register clear command under conditions where FASTAT is set to 10h

45.2.7 Flash Status Register 1 (FSTATR1)

Address(es): 007F FFB1h

	b7	b6	b5	b4	b3	b2	b1	b0
	FCUERR	—	—	FLOCKST	—	—	—	—
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 and cannot be modified.	R
b4	FLOCKST	Lock Bit Status	0: Protected 1: Not protected	R
b6, b5	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7	FCUERR	FCU Error	0: No error occurs in the FCU processing 1: An error occurs in the FCU processing	R

When on-chip ROM is disabled, the data read from FSTATR1 is 00h.
FSTATR1 is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

FLOCKST Bit (Lock Bit Status)

This bit is to reflect the read data of a lock bit when using the lock bit read 2 command.
When the FSTATR0.FRDY bit is set to 1 after a lock bit read 2 command is issued, valid data is stored in the FLOCKST bit. The value of the FLOCKST bit is retained until the completion of the next lock bit read 2 command.

FCUERR Bit (FCU Error)

This bit is used to indicate that an error occurs in the FCU internal processing.
When the FCUERR bit is set to 1, set the FRESETR.FRESET bit to 1 to initialize the FCU. Additionally, recopy the FCU firmware from the FCU firmware area to the FCU RAM area.

45.2.8 Flash Ready Interrupt Enable Register (FRDYIE)

Address(es): 007F C412h

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

Bit	Symbol	Bit Name	Description	R/W
b0	FRDYIE	Flash Ready Interrupt Enable	0: FRDYI interrupt requests disabled 1: FRDYI interrupt requests enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When on-chip ROM is disabled, the data read from FRDYIE is 00h and writing is disabled.

FRDYIE Bit (Flash Ready Interrupt Enable)

This bit is to enable/disable a FRDYI interrupt request when programming/erasure is completed.
If the FRDYIE bit is set to 1, a flash ready interrupt request (FRDYI) is generated when execution of the FCU command has completed (FSTATR0.FRDY bit changes from 0 to 1).

45.2.9 Flash P/E Mode Entry Register (FENTRYR)

Address(es): 007F FFB2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FEKEY[7:0]								FENTRYD	—	—	—	FENTRY3	FENTRY2	FENTRY1	FENTRY0
Value after 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	FENTRY0	ROM P/E Mode Entry 0	0: 512 Kbytes (area 0*2) of ROM are in ROM read mode. 1: 512 Kbytes (area 0*2) of ROM are in ROM P/E mode.	R/W
b1	FENTRY1	ROM P/E Mode Entry 1*1	0: 256 or 512 Kbytes (area 1*3) of ROM are in ROM read mode. 1: 256 or 512 Kbytes (area 1*3) of ROM are in ROM P/E mode.	R/W
b2	FENTRY2	ROM P/E Mode Entry 2*1	0: 512 Kbytes (area 2*4) of ROM are in ROM read mode. 1: 512 Kbytes (area 2*4) of ROM are in ROM P/E mode.	R/W
b3	FENTRY3	ROM P/E Mode Entry 3*1	0: 512 Kbytes (area 3*5) of ROM are in ROM read mode. 1: 512 Kbytes (area 3*5) of ROM are in ROM P/E mode.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	E2 DataFlash P/E Mode Entry	See section 46, E2 DataFlash Memory (Flash Memory for Data Storage).	R/W
b15 to b8	FEKEY[7:0]	Key Code	These bits enable or disable rewriting of the FENTRYD and FENTRY3 to FENTRY0 bits.	R/(W) *6

Note 1. The FENTRY1, FENTRY2, and FENTRY3 bits are provided when the user area capacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively. When the functions of the FENTRY3 to FENTRY1 bits are not provided, these bits are reserved; the description for the originally reserved bits applies to them.
Note 2. Area 0 of 384-Kbyte ROM ranges from addresses 00FA 0000h to 00FF FFFFh for programming/erasure and FFFA 0000h to FFFF FFFFh for reading; and area 0 of 512-Kbyte/768-Kbyte/1-Mbyte/1.5-Mbyte/2-Mbyte ROM ranges from addresses 00F8 0000h to 00FF FFFFh for programming/erasure and FFF8 0000h to FFFF FFFFh for reading.
Note 3. Area 1 of 768-Kbyte ROM ranges from addresses 00F4 0000h to 00F7 FFFFh for programming/erasure and FFF4 0000h to FFF7 FFFFh for reading.
Area 1 of 1-Mbyte/1.5-Mbyte/2-Mbyte ROM ranges from addresses 00F0 0000h to 00F7 FFFFh for programming/erasure and FFF0 0000h to FFF7 FFFFh for reading.
Note 4. Area 2 ranges from addresses 00E8 0000h to 00EF FFFFh for programming/erasure and FFE8 0000h to FFEF FFFFh for reading.
Note 5. Area 3 ranges from addresses 00E0 0000h to 00E7 FFFFh for programming/erasure and FFE0 0000h to FFE7 FFFFh for reading.
Note 6. Write data is not retained.

To place the ROM/E2 DataFlash in ROM P/E mode so that the FCU can accept commands, either the FENTRYD or FENTRY0 to FENTRY3 bits*1 must be set to 1. Note that if a value is set other than AA01h, AA02h*2, AA04h*2, AA08h*2, and AA80h to FENTRYR, the FSTATR0.ILGLERR bit is set to 1 and the FCU enters the command-locked state.

Write access to FENTRYR is valid only when the specific value is written to the upper byte in word access. Any other writing causes the register to be initialized. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from FENTRYR is 0000h and writing is disabled.

FENTRYR is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

Note 1. The FENTRY1, FENTRY2, and FENTRY3 bits are provided when the user area capacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively.
Note 2. AA02h, AA04h, and AA08h can be written to FENTRYR when the user area capacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively.

FENTRY0 Bit (ROM P/E Mode Entry 0)

This bit is used to place 512 Kbytes of ROM (area 0) in ROM P/E mode. Specifically, area 0 of 512-Kbyte ROM ranges from addresses FFF8 0000h to FFFF FFFFh for reading and 00F8 0000h to 00FF FFFFh for programming/erasure.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FSTATR0.FRDY bit is set to 1.
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

- The writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRY0 bit.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRY0 bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

FENTRY1 Bit (ROM P/E Mode Entry 1)

This bit is used to place 256 Kbytes or 512 Kbytes of ROM (area 1) in ROM P/E mode. Specifically, area 1 of 768-Kbyte ROM ranges from addresses FFF4 0000h to FFF7 FFFFh for reading and 00F4 0000h to 00F7 FFFFh for programming/erasure; and area 1 of 1-Mbyte/1.5-Mbyte/2-Mbyte ROM ranges from addresses FFF0 0000h to FFF7 FFFFh for reading and 00F0 0000h to 00F7 FFFFh for programming/erasure. FENTRY1 is provided when the user area capacity exceeds 512 Kbytes.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FSTATR0.FRDY bit is set to 1.
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

- The writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRY1 bit.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRY1 bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

FENTRY2 Bit (ROM P/E Mode Entry 2)

This bit is used to place 512 Kbytes of ROM (area 2) in ROM P/E mode. Specifically, area 2 ranges from addresses FFE8 0000h to FFEF FFFFh for reading and 00E8 0000h to 00EF FFFFh for programming/erasure. FENTRY2 is provided when the user area capacity exceeds 1 Mbyte.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FSTATR0.FRDY bit is set to 1.
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

- The writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRY2 bit.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRY2 bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

FENTRY3 Bit (ROM P/E Mode Entry 3)

This bit is used to place 512 Kbytes of ROM (area 3) in ROM P/E mode. Specifically, area 3 ranges from addresses FFE0 0000h to FFE7 FFFFh for reading and 00E0 0000h to 00E7 FFFFh for programming/erasure. FENTRY3 is provided when the user area capacity exceeds 1.5 Mbytes.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FSTATR0.FRDY bit is set to 1.
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

- The writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRY3 bit.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRY3 bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

45.2.10 Flash Protection Register (FPROTR)

Address(es): 007F FFB4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FPKEY[7:0]								—	—	—	—	—	—	—	FPROTCN
Value after 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	FPROTCN	Lock Bit Protection Cancel	0: Protection with a lock bit enabled 1: Protection with a lock bit disabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FPKEY[7:0]	Key Code	These bits are used to enable or disable rewriting of the FPROTCN bit.	R/(W) *1

Note 1. Write data is not retained.

Write access to FPROTR is valid only when the specific value is written to the upper byte in word access. Any other writing causes the register to be initialized. Data written to the upper byte is not retained.
When on-chip ROM is disabled, the data read from FPROTR is 0000h and writing is disabled.
FPROTR is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

FPROTCN Bit (Lock Bit Protection Cancel)

This bit is used to enable/disable the programming/erasure protection with a lock bit.

[Setting condition]

- 55h is written to the FPKEY[7:0] bits and 1 is written to the FPROTCN bit in word access when the value of FENTRYR is other than 0000h.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FPKEY[7:0] bits are other than 55h.
- 55h is written to the FPKEY[7:0] bits and 0 is written to the FPROTCN bit in word access.
- The value of FENTRYR is 0000h.

45.2.11 Flash Reset Register (FRESETR)

Address(es): 007F FFB6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FRKEY[7:0]								—	—	—	—	—	—	—	FRESET
Value after 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	FRESET	Flash Reset	0: FCU is not reset 1: FCU is reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FRKEY[7:0]	Key Code	These bits are used to enable or disable rewriting of the FRESET bit.	R/(W) *1

Note 1. Write data is not retained.

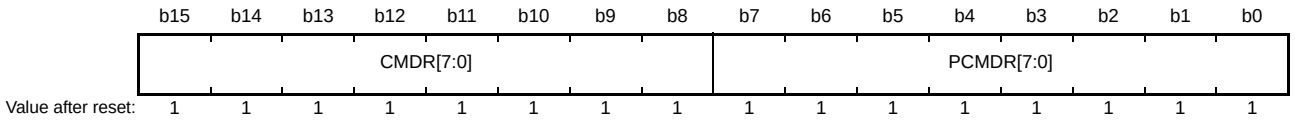
Write access to FRESETR is valid only when the specific value is written to the upper byte in word access. Data written to the upper byte is not retained.
When on-chip ROM is disabled, the data read from FRESETR is 0000h and writing is disabled.

FRESET Bit (Flash Reset)

When the FRESET bit is set to 1, programming/erasure operations for the ROM/E2 DataFlash are forcibly terminated, and the FCU is initialized.
High voltage is applied to the memory of the ROM/E2 DataFlash during programming/erasure. To ensure the time required for dropping the voltage applied to the memory, keep the FRESET bit set to 1 for tRESW2 (see section 49, Electrical Characteristics) when initializing the FCU. While the FRESET bit is kept 1, prohibit the ROM/E2 DataFlash from being read. Additionally, when the FRESET bit is set to 1, the FCU commands cannot be used because FENTRYR is initialized.
Writing of the FRESET bit is enabled only in word access and when the FRKEY[7:0] bits are CCh.

45.2.12 FCU Command Register (FCMDR)

Address(es): 007F FFBAh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCMDR[7:0]	Precommand	Store the command immediately before the last command received by the FCU.	R
b15 to b8	CMDR[7:0]	Command	Store the last command received by the FCU.	R

When on-chip ROM is disabled, data read from FCMDR is 0000h and writing is disabled.
FCMDR is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.
Table 45.4 lists the states of FCMDR after receiving each command. For details on the blank check processing, see section 46.6, Programming and Erasing the E2 DataFlash Memory.

Table 45.4 States of FCMDR after Receiving Each Command

Command	CMDR[7:0]	PCMDR[7:0]
P/E normal mode transition	FFh	Previous command
Status read mode transition	70h	Previous command
Lock bit read mode transition (lock bit read 1)	71h	Previous command
Peripheral clock notification	E9h	Previous command
Programming	E8h	Previous command
Block erase	D0h	20h
P/E suspend	B0h	Previous command
P/E resume	D0h	Previous command
Status register clear	50h	Previous command
Lock bit read 2/blank check	D0h	71h
Lock bit programming	D0h	77h

45.2.13 FCU Processing Switching Register (FCPSR)

Address(es): 007F FFC8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESUSPMD
Value after 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	ESUSPMD	Erasure Suspend Mode	0: Suspension priority mode 1: Erasure priority mode	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When on-chip ROM is disabled, the data read from FCPSR is 0000h and writing is disabled.
FCPSR is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

ESUSPMD Bit (Erasure Suspend Mode)

This bit is to select the erasure suspend mode for when a P/E suspend command is issued while the FCU executes the erasure processing for the ROM/E2 DataFlash. For details, see section 45.7, Suspending Operation.

45.2.14 Flash P/E Status Register (FPESTAT)

Address(es): 007F FFCCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PEERRST[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	PEERRST[7:0]	P/E Error Status	00h: No programming error 01h: Programming error against areas protected by a lock bit 02h: Programming error due to sources other than the lock bit protection 11h: Erasure error against areas protected by a lock bit 12h: Erasure error due to sources other than the lock bit protection (Values other than above are reserved)	R
b15 to b8	—	Reserved	These bits are read as 0 and cannot be modified.	R

When on-chip ROM is disabled, the data read from FPESTAT is 0000h and writing is disabled.
FPESTAT is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

PEERRST[7:0] Bits (P/E Error Status)

These bits are used to indicate the reason of an error that occurs during the programming/erasure processing for the ROM/E2 DataFlash.
The value of the PEERRST[7:0] bits is valid only when the FSTATR0.FRDY bit is set to 1 while the FSTATR0.ERSERR bit or FSTATR0.PRGERR bit is 1. The value of the reason of the past error is retained in the PEERRST[7:0] bits when the ERSERR bit and PRGERR bit is 0.

45.2.15 Peripheral Clock Notification Register (PCKAR)

Address(es): 007F FFE8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PCKA[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	PCKA[7:0]	Peripheral Clock Notification	These bits are used to set the flashIF clock (FCLK) at the programming/erasure for the ROM/E2 DataFlash.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When on-chip ROM is disabled, the data read from PCKAR is 0000h and writing is disabled.
PCKAR is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

PCKA[7:0] Bits (Peripheral Clock Notification)

These bits are used to set the flashIF clock (FCLK) at the programming/erasure for the ROM/E2 DataFlash.
Set the FCLK frequency in the PCKA[7:0] bits and issue a peripheral clock notification command before programming/erasure. Do not change frequency during programming/erasure for the ROM/E2 DataFlash.
Calculate the setting value as follows:

- Convert an operating frequency represented in MHz units to binary and write it to the PCKA[7:0] bits.

For example, when the operating frequency of the flashIF clock is 35.9 MHz, the setting value is calculated as follows:

- Round 35.9 off to a whole number.
- Convert 36 to binary and set the upper bits and lower bits of the PCKA[7:0] bits to 00h and 24h (0010 0100b).

Note: • When the PCKA[7:0] bits are set to values outside the range from 4 to 50 MHz, do not issue a programming command to the ROM/E2 DataFlash.
Note: • When the PCKA[7:0] bits are set to a frequency that is different from the FCLK, the data of the ROM/E2 DataFlash may be damaged.
Note: • Please note that the programming time depends on the frequency to some extent even if the PCKA[7:0] bits are used.

45.3 Configuration of Memory Areas for the ROM

The ROM of products in the RX63N/RX631 Group is configured of a maximum 2-Mbyte user area. The address range occupied by this area is shown in Figure 45.2.
Note that for the user area, the address range for reading differs from the address range for programming and erasure.

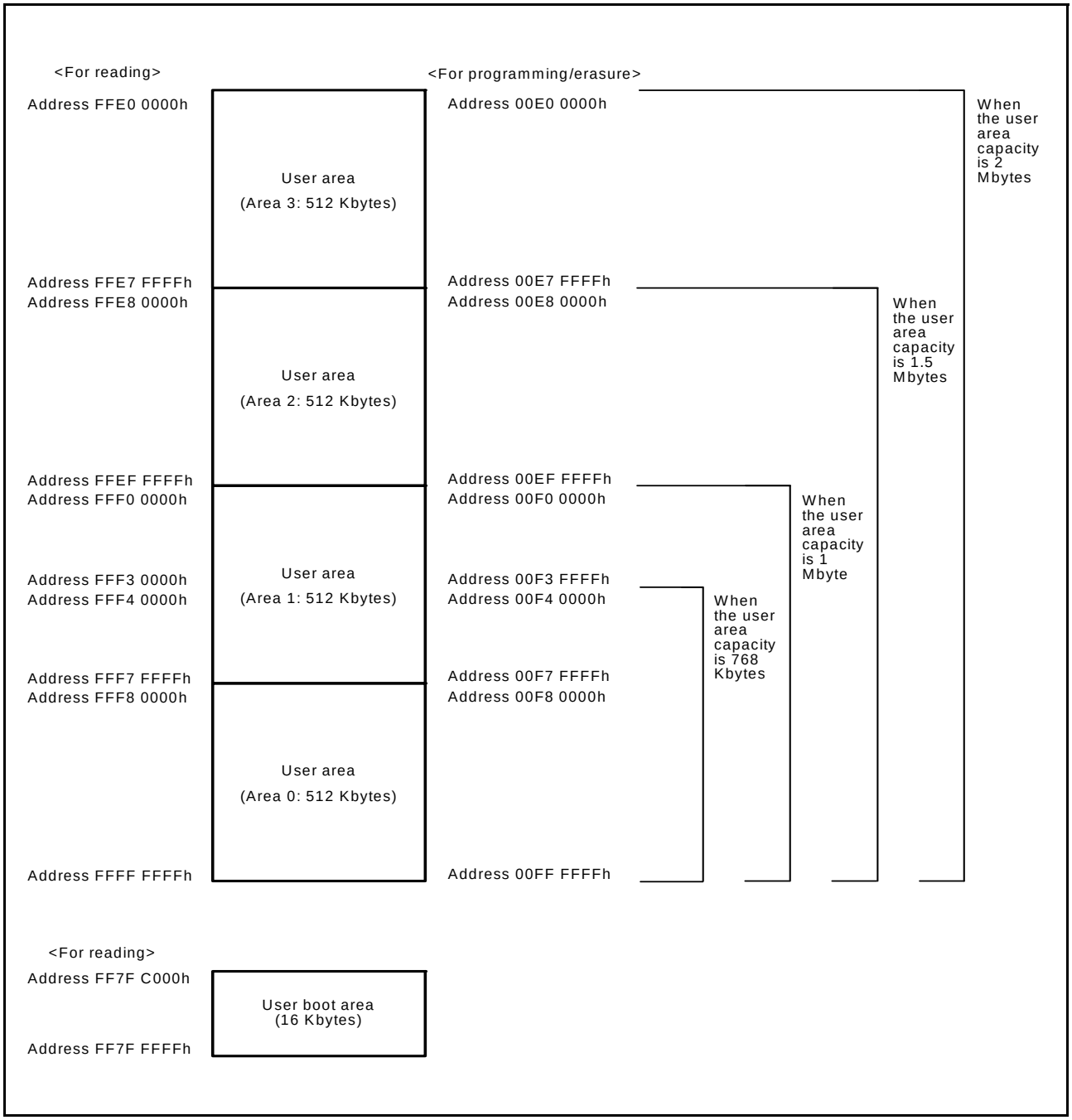


Figure 45.2 Memory Area Configuration of ROM

45.4 Block Configuration

The configuration of erasure blocks for the user area is shown in Figure 45.3.
The user area is divided into following blocks and erasure is executed in block units. Programming is executed in 128-byte units, where each unit starts at 00h.

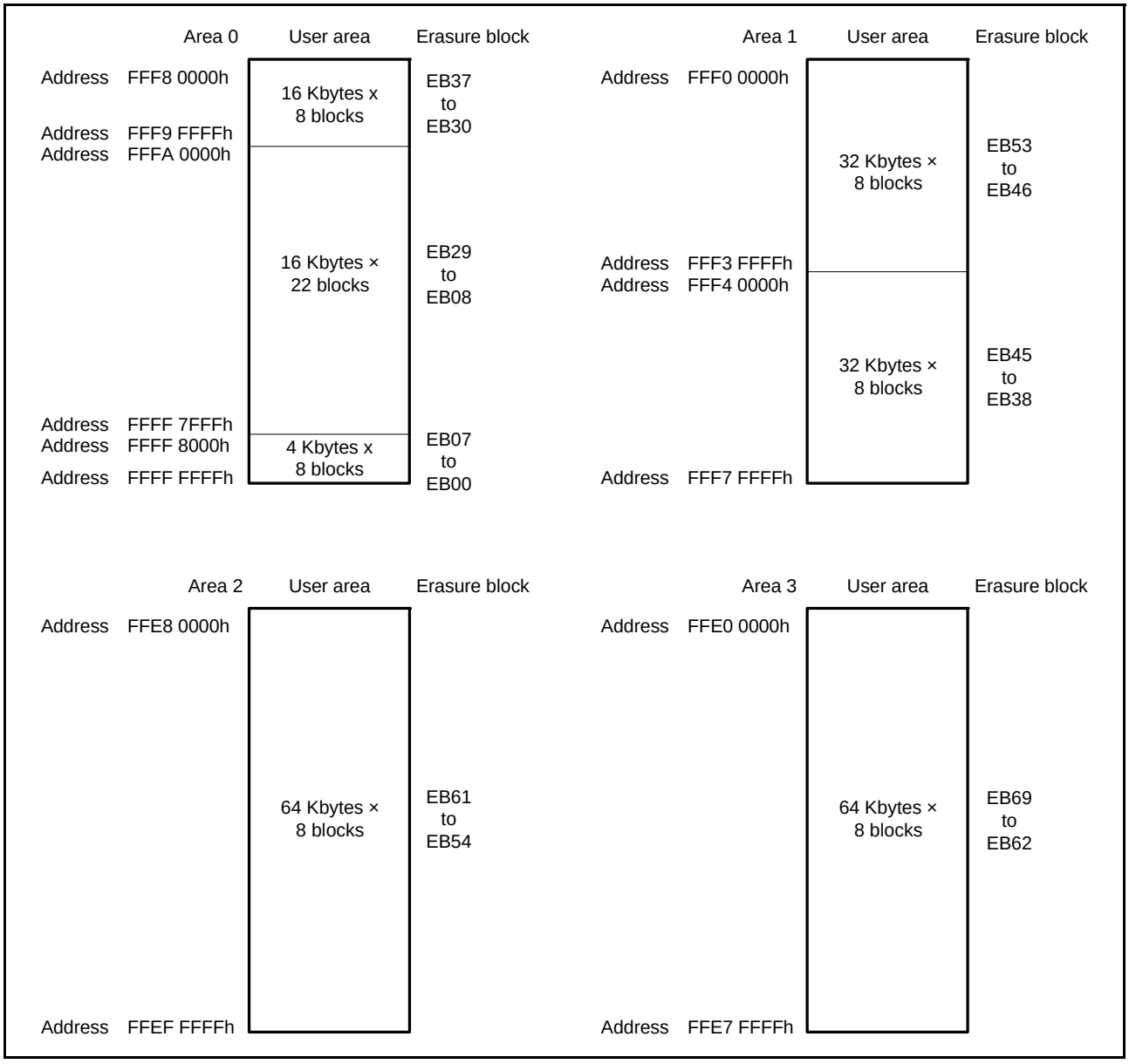


Figure 45.3 Configuration of Erasure Blocks for the User Area

45.5 Operating Modes Associated with the ROM

For information on the relationship between the setting of the level on the MD pin and the operating mode for the RX63N/RX631 Group, refer to section 3, Operating Modes.
The ROM can be read, programmed, and erased on board in boot mode, USB boot mode, user boot mode, single-chip mode (with on-chip ROM enabled), or on-chip-ROM-enabled expansion mode.
Which areas are programmable and erasable, the area from which booting up proceeds after a reset, etc., differs with the mode. The differences between modes are indicated in Table 45.5.

Table 45.5 Differences between Modes

Item	Boot Mode	USB Boot Mode/User Boot Mode	Single-Chip Mode (with On-Chip ROM Enabled) or On-Chip-ROM-Enabled Expansion Mode
Environment for programming and erasure	On-board programming		
Programmable and erasable area	User area/user boot area/ data area	User area/data area	User area/data area
Division into erasure blocks	Possible*1	Possible*1	Possible
Boot program at a reset	Boot program	User boot program	User program

Note 1. The entire ROM may be erased at the time of booting up. Specified blocks can subsequently be erased. For details, refer to section 45.10.4, ID Code Protection (Boot Mode), section 45.10.2, State Transitions in Boot Mode, and section 45.11.2, State Transitions.

- Programming and erasure of the user boot area are only possible in boot mode.
- In boot mode, a host is able to program, erase, or read the user area, user boot area, or data area via an SCI.
- In boot mode, on-chip RAM is employed for the embedded program for use in boot mode. For this reason, preserving the contents of on-chip RAM is not possible in this case.
- Booting-up in USB boot mode and user boot mode is from the user boot area. The user boot area of the product as-shipped holds the USB boot program, which is capable of reading from or writing to the user area and data area. Furthermore, rewriting of the user boot area in boot mode can enable reading from or writing to the user area and data area via any desired interface.

45.6 Programming and Erasing the ROM

The ROM is programmed and erased by issuing commands to a dedicated sequencer (FCU) for programming and erasure. The FCU has five modes. For programming and erasure, the mode is changed and then commands for programming and erasure are issued.
The mode transitions required to program or erase the ROM and the system of commands are described below. The descriptions apply in common to boot mode, USB boot mode, user boot mode, single-chip mode (with on-chip ROM enabled), and on-chip ROM enabled expansion mode.

45.6.1 FCU Modes

The FCU has five modes. Transitions between modes are caused by modifying FENTRYR or issuing FCU commands. Figure 45.4 is a diagram of the FCU mode transitions.

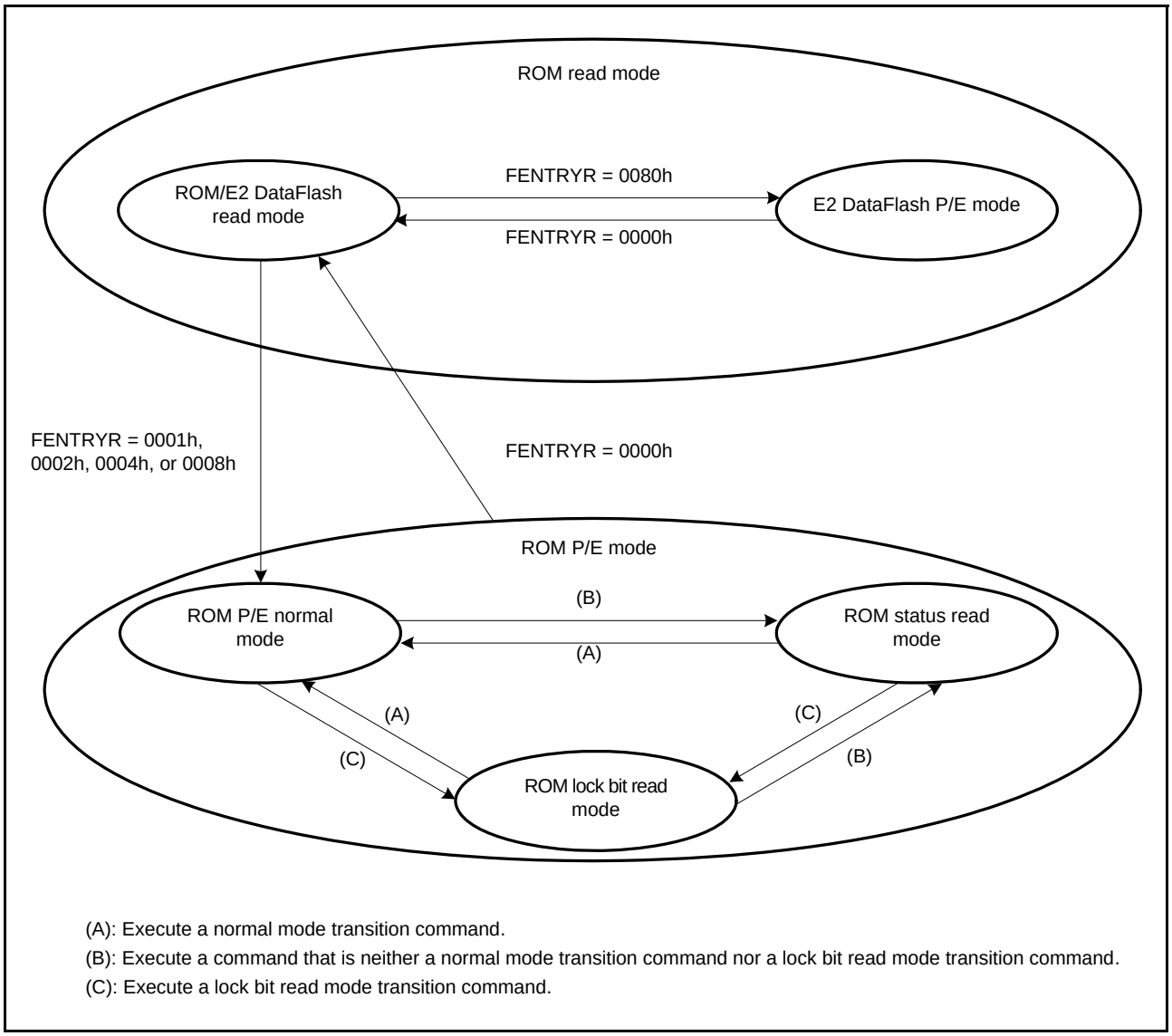


Figure 45.4 Mode Transitions of the FCU (Associated with the ROM)

45.6.1.1 ROM Read Modes

The ROM read modes are for high-speed reading of the ROM. Access to an address for reading can be accomplished in one cycle of ICLK.
ROM/E2 DataFlash read mode and E2 DataFlash P/E mode are the two kinds of ROM read modes.

(1) ROM/E2 DataFlash Read Mode

This mode is for reading the ROM and E2 DataFlash memory. The FCU does not accept FCU commands. The FCU enters this mode when the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits are set to 0 with the FENTRYR.FENTRYD bit set to 0.

(2) E2 DataFlash P/E Modes

These modes are for programming and erasure of the E2 DataFlash memory. High-speed reading of the ROM is also possible. Although the FCU accepts FCU commands related to the E2 DataFlash memory in this mode, it does not accept FCU commands related to the ROM. The FCU enters these modes when the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits are set to 0 with the FENTRYR.FENTRYD bit set to 1.
For details on the E2 DataFlash P/E modes, see section 46.6.1, FCU Modes (in section 46, E2 DataFlash Memory (Flash Memory for Data Storage)).

45.6.1.2 ROM P/E Modes

The ROM P/E modes are for programming and erasure of the ROM. High-speed reading of the ROM is not possible in these modes. Read access to an address within the range for reading causes a ROM-access violation, and the FCU enters the command-locked state (see section 45.8.2, Error Protection).
ROM P/E normal mode, ROM status read mode, and ROM lock-bit read mode are the three ROM P/E modes.

(1) ROM P/E Normal Mode

The transition to ROM P/E normal mode is the first transition in the process of programming or erasing the ROM. The FCU enters this mode when the FENTRYR.FENTRYD bit is set to 0, with any of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits set to 1 in ROM read mode, or when the normal mode transition command is received in ROM P/E modes. Table 45.8 lists the acceptable commands in this mode.
Read access to an address within the range for programming and erasure while any of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits are set to 1 causes a ROM-access violation, and the FCU enters the command-locked state (see section 45.8.2, Error Protection).

(2) ROM Status Read Mode

In the ROM status read mode, the state of the ROM can be read. The FCU enters this mode when a command other than the normal mode transition or lock bit read mode transition command is received in ROM P/E modes.
ROM status read mode encompasses the states where the FSTATR0.FRDY bit is 0 and the command-locked state after an error has occurred. Table 45.8 lists the acceptable commands in this mode.
Read access to an address within the range for programming and erasure while any of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits are 1 allows the value of FSTATR0 to be read.

(3) ROM Lock-Bit Read Mode

In the ROM lock-bit read mode, reading the ROM allows the lock bits to be read. The FCU enters this mode when a lock-bit read mode transition command is received in ROM P/E modes. Table 45.8 lists the acceptable commands in this mode. Read access to an address within the range for programming and erasure while any of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits are 1 allows the value of the lock bit of the erasure block including the accessed address to be read from all the read bits.

45.6.2 FCU Commands

FCU commands consist of commands for mode transitions of the FCU and commands for programming and erasure. Table 45.6 lists the FCU commands for use with the ROM.

Table 45.6 FCU Commands for Use with the ROM

Command	Description
P/E normal mode transition	Shifts to normal mode (see section 45.6.3, Connections between FCU Modes and Commands)
Status read mode transition	Shifts to status read mode (see section 45.6.3, Connections between FCU Modes and Commands)
Lock bit read mode transition (lock bit read 1)	Shifts to lock bit read mode (see section 45.6.3, Connections between FCU Modes and Commands)
Peripheral clock notification	Sets the frequency of the peripheral clock
Programming	ROM programming (in 128-byte units)
Block erase	ROM erasure (in block units, with the lock bit being erased simultaneously)
P/E suspend	Suspends programming/erasure
P/E resume	Resumes programming/erasure
Status register clear	Clears the ILGLERR, ERSERR and PRGERR bits in FSTATR0 and releases the FCU from the command locked state
Lock bit read 2/blank check	Reads the lock bit of a specified erasure block (the value of the lock bit is reflected in the FSTATR1.FLOCKST bit)/blank checking of the E2 DataFlash memory
Lock bit programming	Programs the lock bit of a specified erasure block

The lock bit read 2 command is also used as the blank check command for the E2 DataFlash memory. That is, when a lock bit read 2 command is issued for the E2 DataFlash, blank checking is executed for the E2 DataFlash memory (see section 46, E2 DataFlash Memory (Flash Memory for Data Storage)).

Commands for the FCU are issued by write access to addresses within the range for ROM programming and erasure. Table 45.7 lists the formats of the FCU commands. Write access as listed in Table 45.7 and in accordance with certain conditions causes the FCU to execute processing for the corresponding command.

For details on the conditions for the acceptance of the individual FCU commands, see section 45.6.3, Connections between FCU Modes and Commands. For how to use the FCU commands, see section 45.6.4, FCU Command Usage.

Table 45.7 FCU Command Formats

Command	Number of Bus Cycles	1st Cycle		2nd Cycle		3rd Cycle		4th to 5th Cycles		6th Cycle		7th to 66th Cycles		67st Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
P/E normal mode transition	1	RA	FFh	—	—	—	—	—	—	—	—	—	—	—	—
Status read mode transition	1	RA	70h	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read mode transition (lock bit read 1)	1	RA	71h	—	—	—	—	—	—	—	—	—	—	—	—
Peripheral clock notification	6	RA	E9h	RA	03h	RA	0F0Fh	RA	0F0Fh	RA	D0h	—	—	—	—
Programming	67	RA	E8h	RA	40h	WA	WDn	RA	WDn	RA	WDn	RA	WDn	RA	D0h
Block erase	2	RA	20h	BA	D0h	—	—	—	—	—	—	—	—	—	—
P/E suspend	1	RA	B0h	—	—	—	—	—	—	—	—	—	—	—	—
P/E resume	1	RA	D0h	—	—	—	—	—	—	—	—	—	—	—	—
Status register clear	1	RA	50h	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read 2	2	RA	71h	BA	D0h	—	—	—	—	—	—	—	—	—	—
Lock bit programming	2	RA	77h	BA	D0h	—	—	—	—	—	—	—	—	—	—

Address column RA: ROM programming/erasure address
When the FENTRYR.FENTRY0 bit is 1: An address from 00F8 0000h to 00FF FFFFh
When the FENTRYR.FENTRY1 bit is 1 with 768-Kbyte ROM: An address from 00F4 0000h to 00F7 FFFFh
When the FENTRYR.FENTRY1 bit is 1 with 1-Mbyte/1.5-Mbyte/2-Mbyte ROM: An address from 00F0 0000h to 00F7 FFFFh
When the FENTRYR.FENTRY2 bit is 1: An address from 00E8 0000h to 00EF FFFFh
When the FENTRYR.FENTRY3 bit is 1: An address from 00E0 0000h to 00E7 FFFFh
WA: ROM programming-destination address
Start address for programming of 128 bytes of data
BA: ROM erasure block address
An address within the target erasure block (specified as an address in the range for programming and erasure)

Data column WDn: nth word of data for programming (n = 1 to 64)

45.6.3 Connections between FCU Modes and Commands

The sets of FCU commands that can be accepted in each of the FCU modes are fixed. Furthermore, which commands are acceptable in a given FCU mode also depends on the state of the FCU.
Issuing of an FCU command must follow checking of the FCU's state after transitions of the FCU mode.
Commands that are acceptable in the various FCU modes and states are listed in Table 45.8. Issuing a command that is not currently acceptable leads to the FCU being placed in the command-locked state (see section 45.8.2, Error Protection).
Issuing of an FCU command must follow checking of the values of the FRDY, ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and of the FSTATR1.FCUERR bit after transitions of the FCU mode. Furthermore, the FASTAT.CMDLK bit can be checked to see if an error has occurred. The value of the FASTAT.CMDLK bit is the logical OR of the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FSTATR1.FCUERR bit.

Table 45.8 Acceptable Commands and the State and Mode (ROM P/E Mode) of the FCU

	P/E Normal Mode			**Status Read Mode**									**Lock-Bit Read Mode**		
	Programming Suspended	**Erasure Suspended**	**Other State**	**Programming or Erasure**	**Programming while erasure is suspended**	**Processing to Suspend Programming or Erasure**	**Lock Bit Read 2 Processing**	**Programming Suspended**	**Erasure Suspended**	**Command-Locked State (FRDY = 0)**	**Command-Locked State (FRDY = 1)**	**Other State**	**Programming Suspended**	**Erasure Suspended**	**Other State**
FSTATR0.FRDY bit	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0/1**	**1**	**1**	**1**	**1**	**1**
FSTATR0.SUSRDY bit	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
FSTATR0.ERSSPD bit	**0**	**1**	**0**	**0**	**1**	**0/1**	**0**	**0**	**1**	**0/1**	**0/1**	**0**	**0**	**1**	**0**
FSTATR0.PRGSPD bit	**1**	**0**	**0**	**0**	**0**	**0/1**	**0**	**1**	**0**	**0/1**	**0/1**	**0**	**1**	**0**	**0**
FASTAT.CMDLK bit	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
Normal mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Status read transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Lock-bit read transition (lock bit read 1)	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Peripheral clock notification	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A
Programming	×	*	A	×	×	×	×	×	*	×	×	A	×	*	A
Block erase	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A
P/E suspend	×	×	×	A	×	×	×	×	×	×	×	×	×	×	×
P/E resume	A	A	×	×	×	×	×	A	A	×	×	×	A	A	×
Status register clear	A	A	A	×	×	×	×	A	A	A	A	A	A	A	A
Lock bit read 2	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Lock bit programming	×	*	A	×	×	×	×	×	*	×	×	A	×	*	A

A: Acceptable
*: Only programming is acceptable for blocks other than the block where erasure was suspended
×: Not acceptable

45.6.4 FCU Command Usage

The set of FCU commands consists of commands for FCU mode transitions, actually programming or erasing the ROM, error processing, and suspension and resumption. The following passages describe the various commands. For a description of the modes and states where the respective commands are acceptable, see section 45.6.3, Connections between FCU Modes and Commands.

45.6.4.1 Mode Transitions

This subsection covers the commands for mode transitions. For an illustration of the various transitions between modes, see Figure 45.4.

(1) Switching to ROM P/E Mode

A transition to ROM P/E mode is required before executing an FCU command for the ROM becomes possible.
Setting any of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits to 1 causes a transition to ROM P/E mode for programming and erasure of the corresponding address range.
Before actually proceeding to program or erase the ROM, enable programming and erasure by writing 01h as a byte to FWEPROR (see section 45.2.1, Flash Write Erase Protection Register (FWEPROR)).

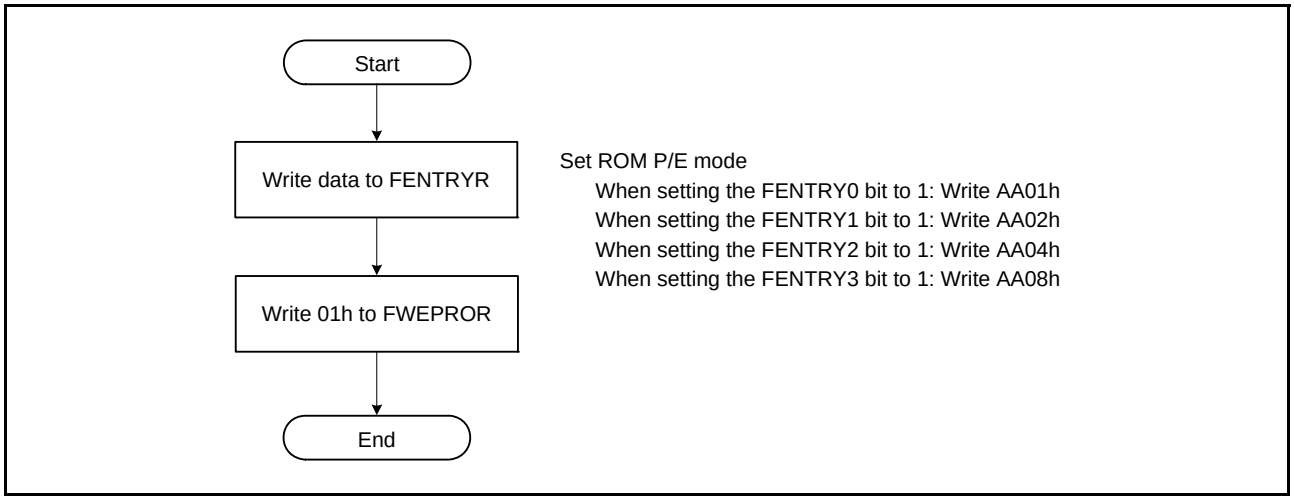
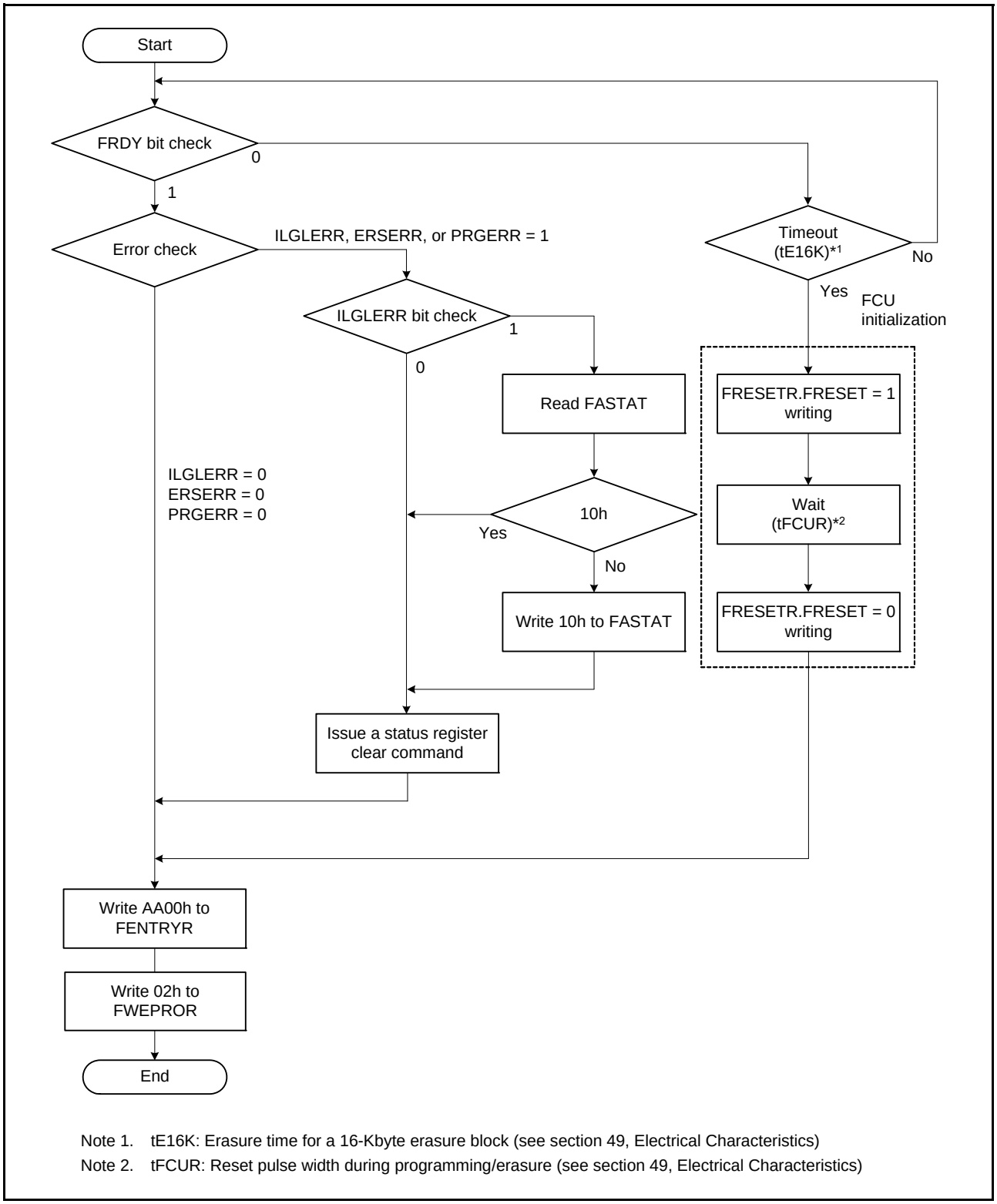


Figure 45.5 Procedure for Transition to ROM P/E Mode

(2) Switching to ROM Read Mode

High-speed reading of the ROM requires clearing of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits to 0000b, which places the FCU in ROM read mode.
Writing of 02h as a byte to FWEPROR is also required to disable programming and erasure (see section 45.2.1, Flash Write Erase Protection Register (FWEPROR)).
Before switching the FCU from ROM P/E mode to read mode, ensure that all processing of FCU commands has been completed and that the FCU has not detected an error.



Note 1. tE16K: Erasure time for a 16-Kbyte erasure block (see section 49, Electrical Characteristics)
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

Figure 45.6 Procedure for Transition to ROM Read Mode

(3) Switching to ROM P/E Normal Mode

Two methods are available for the transition to ROM P/E normal mode: setting FENTRYR while the FCU is in ROM read mode (see section 45.6.1, FCU Modes), or issuing the normal mode transition command while the FCU is in ROM P/E mode (see Figure 45.7). The normal mode transition command is issued by writing FFh as a byte to a ROM programming/erasure address.

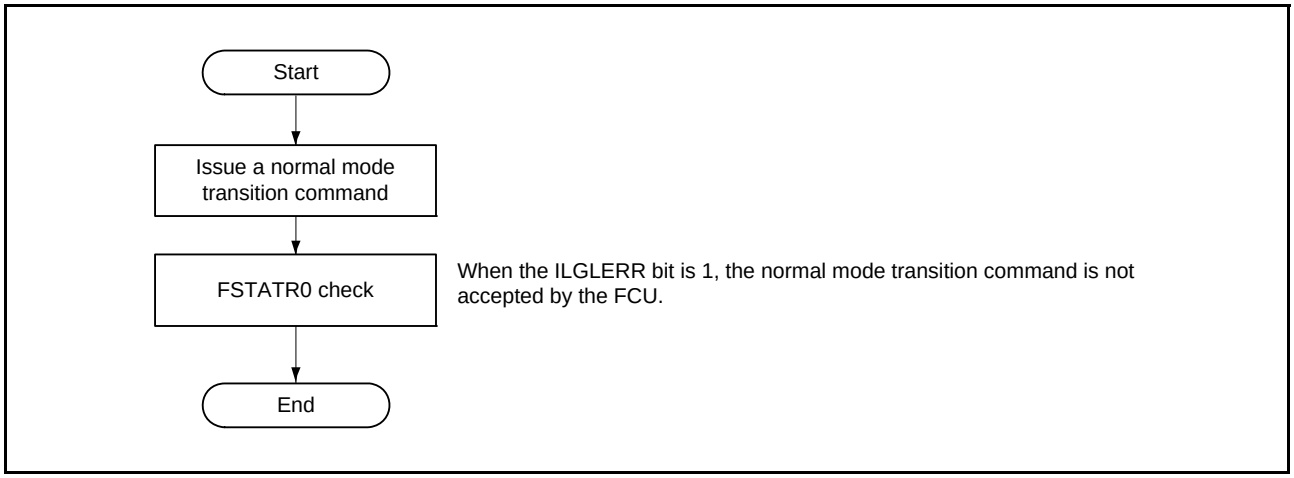


Figure 45.7 Procedure for Transition to ROM P/E Normal Mode

(4) Switching to ROM Status Read Mode

Issuing an FCU command other than a normal mode transition or lock bit read mode transition command places the FCU in ROM status read mode. The same transition can be obtained by issuing the status read mode transition command. Figure 45.8 shows the procedure for checking FSTATR0 as an example. In the example, the status read mode transition command is issued to place the FCU in ROM status read mode, and the value of FSTATR0 is obtained by read access to a ROM programming/erasure address and then checked.

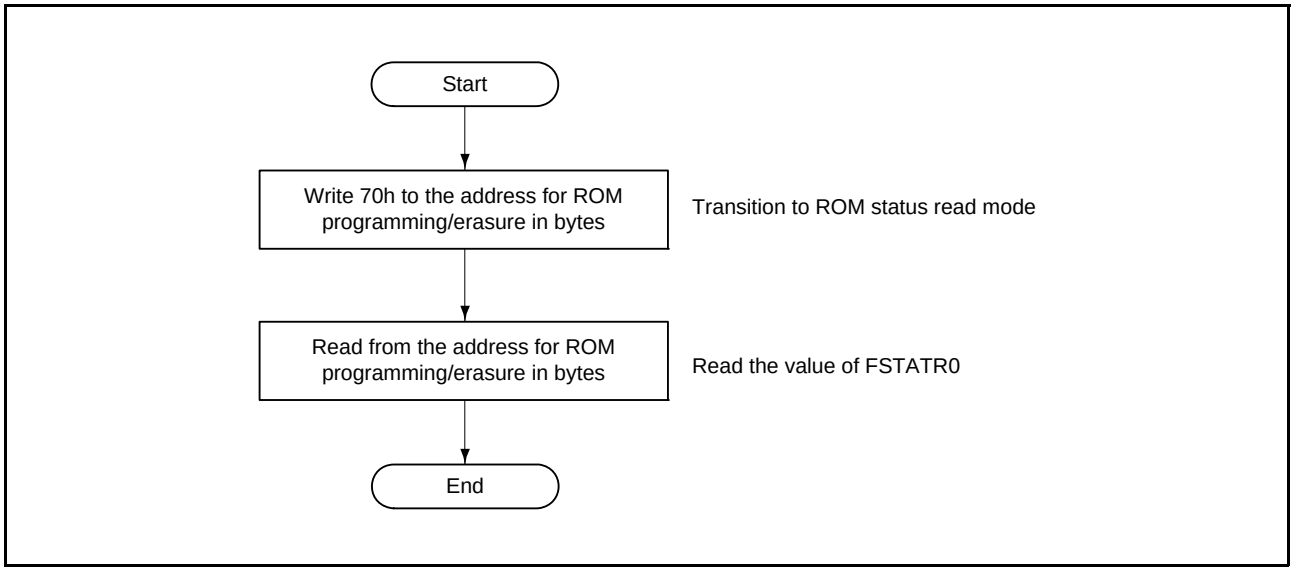


Figure 45.8 Procedure for Transition to ROM Status Read Mode and the Status Checking

(5) Switching to ROM Lock-Bit Read Mode

Clearing the FMODR.FRDMD bit (memory area method) issues a lock bit read mode transition (lock bit read 1) command. After the transition to ROM lock bit read mode, the lock bit value is obtained by read access to a ROM programming/erasure address. All bits of a value thus read have the value of the lock bit of the erasure block that contains the accessed address (Figure 45.9).

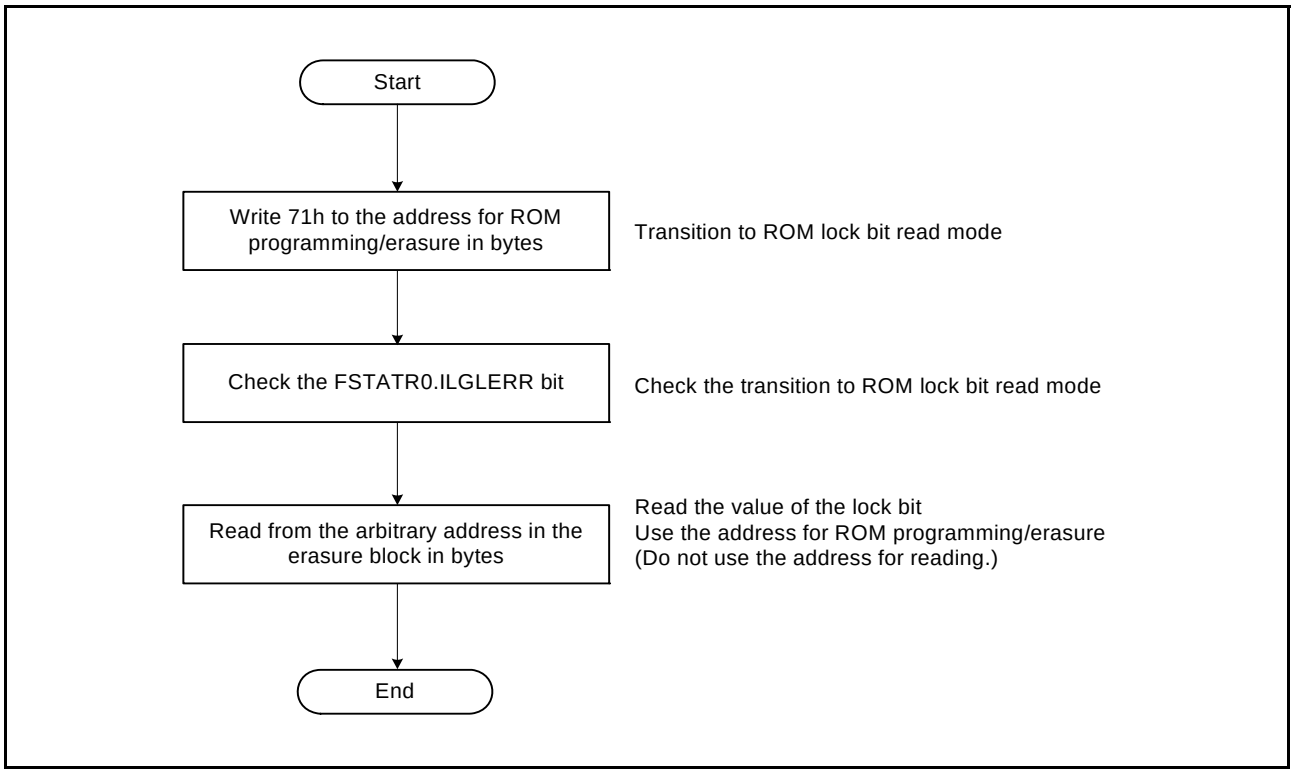
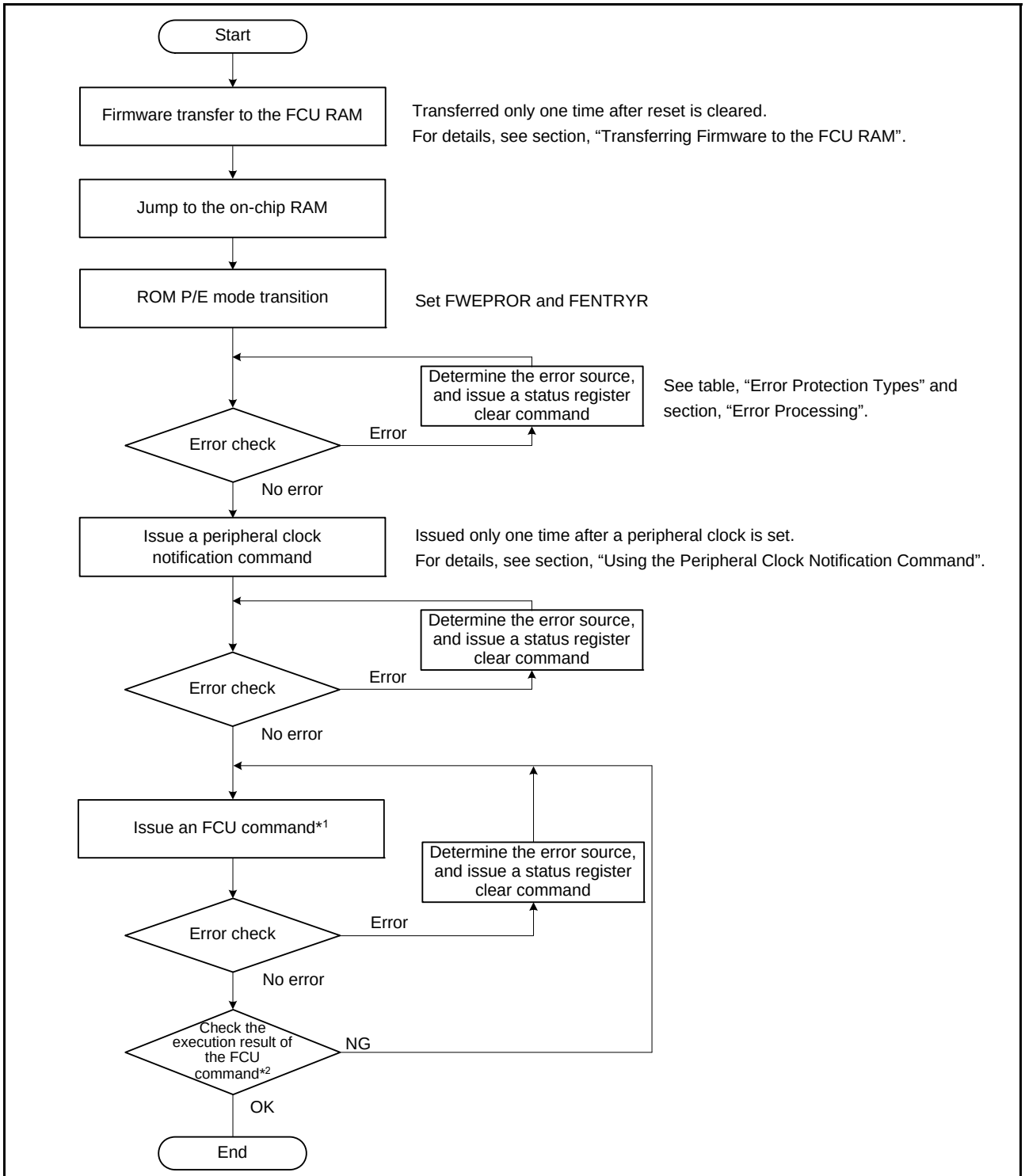


Figure 45.9 Procedure for Transition to ROM Lock-Bit Read Mode and Lock-Bit Reading

45.6.4.2 Programming and Erasure Procedures

The following passages describe the flow of procedures for programming or erasing the ROM. For details on the acceptance of commands by the FCU, see section 45.6.3, Connections between FCU Modes and Commands. Figure 45.10 is a simple flowchart of the procedure for executing FCU commands.



Note 1. Selected among the programming, block erase, lock bit programming, and lock bit read 2 commands.
Note 2. To check the result of programming/erasure processing, enter ROM read mode and read data from the ROM (see section, “Switching to ROM Read Mode”).

Figure 45.10 Simple Flowchart of the Procedure for Programming and Erasure

(1) Transferring Firmware to the FCU RAM

FCU commands can only be used if the FCU RAM holds the firmware for the FCU. The FCU RAM does not hold the FCU firmware immediately after the chip has been booted up, so the firmware must be copied from the FCU firmware area to the FCU RAM. Furthermore, when the FSTATR1.FCUERR bit is set to 1, the FCU must be reset and the firmware recopied because the firmware stored in the FCU RAM may have been corrupted.
Figure 45.11 shows the flow of the procedure for transferring firmware to the FCU RAM. Before writing data to the FCU RAM, set FENTRYR to 0000h and stop the FCU. See section 17, DMA Controller (DMACA), for information on setting up the DMAC to handle data transfer.

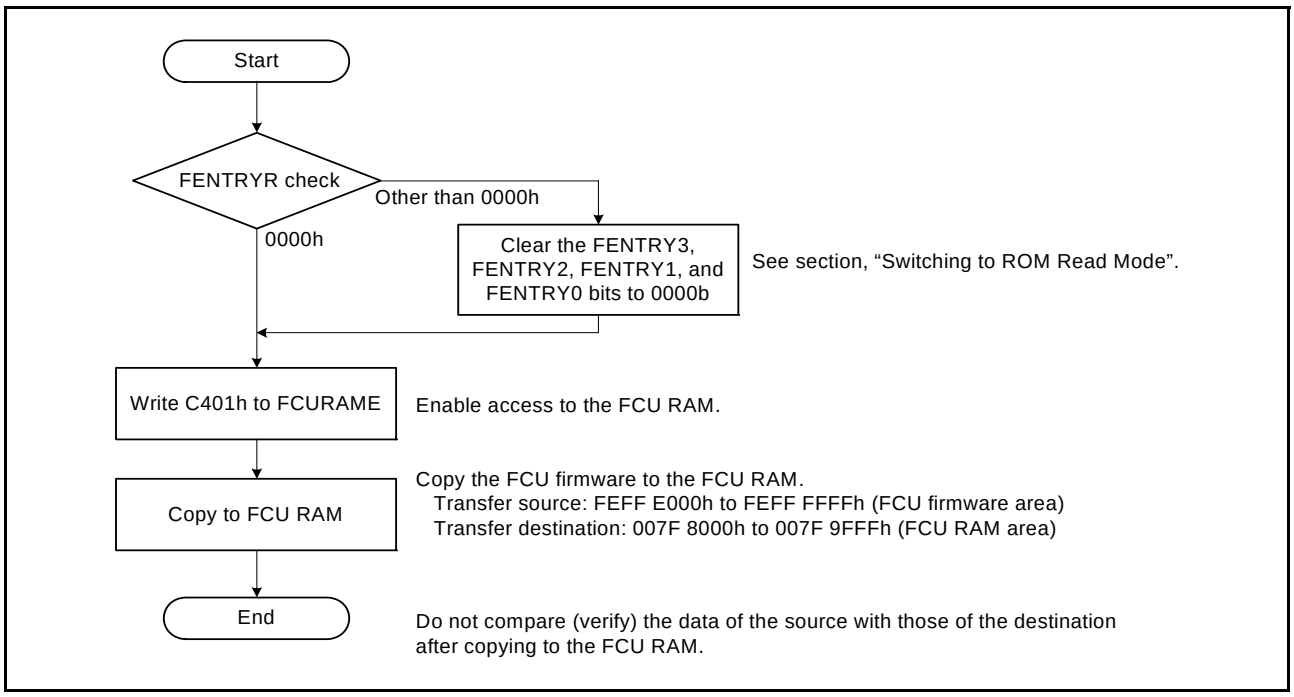


Figure 45.11 Procedure for Firmware Transfer to FCU RAM

(2) Jumping to Locations in On-chip RAM

Since fetching instructions from the ROM is not possible while the ROM is being programmed or erased, code has to be executed from an area other than the ROM. Copy the required instruction code to on-chip RAM and then make a jump to the address where the code starts in on-chip RAM.

(3) Transition to ROM P/E Mode

The FCU is placed in ROM P/E mode by setting the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits and FWEPROR register. For details, see section 45.6.4.1, (3) Switching to ROM P/E Normal Mode.

(4) Using the Peripheral Clock Notification Command

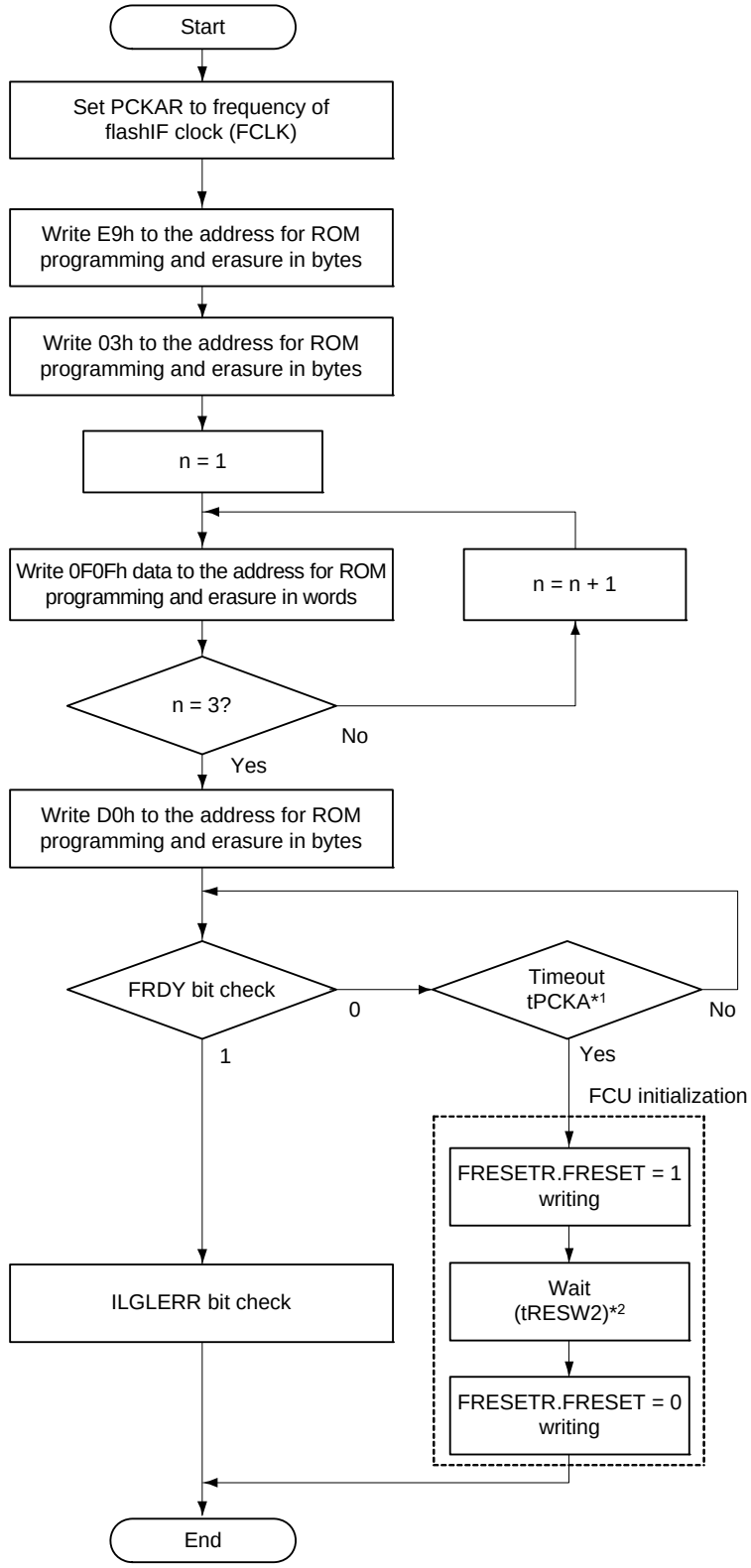
The flashIF clock (FCLK) is used in programming and erasing the ROM, so the frequency of this clock has to be set in PCKAR. Frequencies in the range from 1 to 100 MHz are selectable. If a frequency within this range has not been set, the FCU will detect the error and enter the command-blocked state (see section 45.8.2, Error Protection).

Note that if the PCKA[7:0] bits in the PCKAR register are set to values outside the range from 4 MHz to 50 MHz, do not issue a programming command to the ROM/E2 DataFlash.

The peripheral clock notification command is used after the PCKAR setting has been made. In the first and second cycles for the peripheral clock notification command, respectively, the values E9h and 03h are written as a byte to the address range for programming and erasure of the ROM. Word-size writing is used in the third to fifth cycles of the command. Accordingly, make sure that the addresses used are aligned with four-byte boundaries. After 0F0Fh has been written three times (as a word) to the address range for programming and erasure of the ROM, the process of the FCU setting the frequency of the peripheral clock starts once the value D0h has been written as a byte in the sixth cycle. The FSTATR0.FRDY bit can be used to check whether or not the settings have been completed.

Addresses that can be used in the first to sixth cycles differ according to the settings of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits. Ensure that the addresses suit the settings of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits. If issuing of the command is attempted with an erroneous combination of the settings of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits and the specified addresses, the FCU will detect the error and enter the command-blocked state (see section 45.8.2, Error Protection).

Furthermore, if the setting for the peripheral clock in use will not be changed from this setting after release from the reset state, this setting is also valid for the next FCU command.



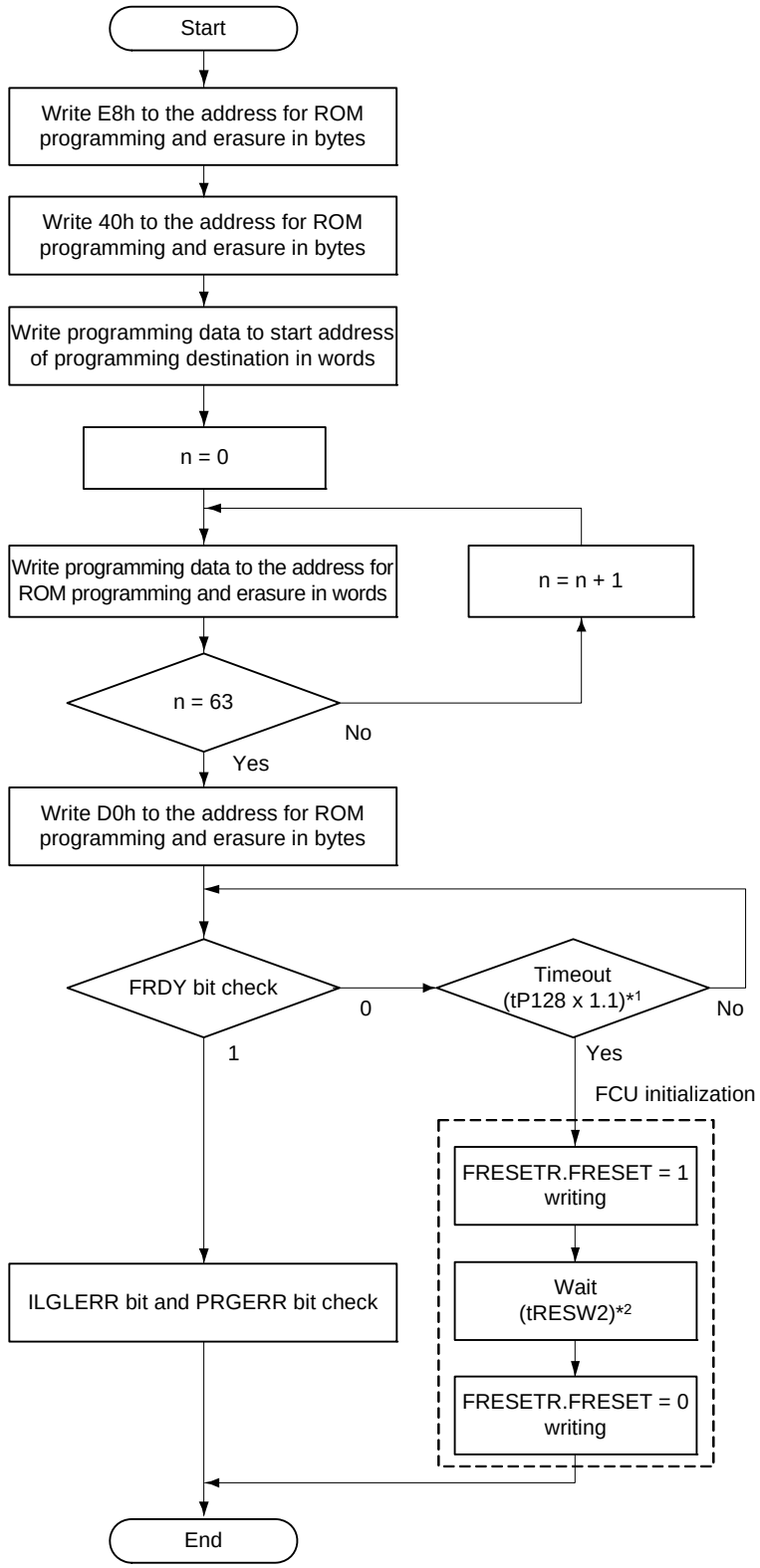
Note 1. tPCKA: 60 µs when FCKL = 50 MHz, 120 µs when FCKL = 25 MHz

Note 2. tRESW2: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

Figure 45.12 Using the Peripheral Clock Notification Command

(5) Programming

The programming command is used to write data to the ROM.
In the first and second cycles for the programming command, respectively, the values E8h and 40h are written as a byte to the address range for programming and erasure of the ROM. In the third cycle, write the actual data to be programmed, as a word unit, to the start address of the target area for programming. For this start address, always use an address that is aligned on a 128-byte boundary. In the fourth to the 66th cycles, write the data for programming in 63 word-unit rounds to the address range for programming and erasure of the ROM. Once the value D0h has been written as a byte to the address range for programming and erasure of the ROM in the 67th cycle, the FCU begins the actual process of programming the ROM. The FSTATR0.FRDY bit can be used to check whether or not the programming has been completed.
Addresses that can be used in the first to 67th cycles differ according to the setting of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits. Ensure that the addresses suit the setting of FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits. If issuing of the command is attempted with an erroneous combination of the setting of the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits and the specified addresses, the FCU will detect the error and enter the command-blocked state (see section 45.8.2, Error Protection).
In cases where the target range in the third to 66th cycles includes addresses that do not require programming, use FFFFh as the data for programming to those addresses. To execute programming with lock bit protection disabled, proceed with programming after setting the FPROTR.FPROTCN bit to 1.

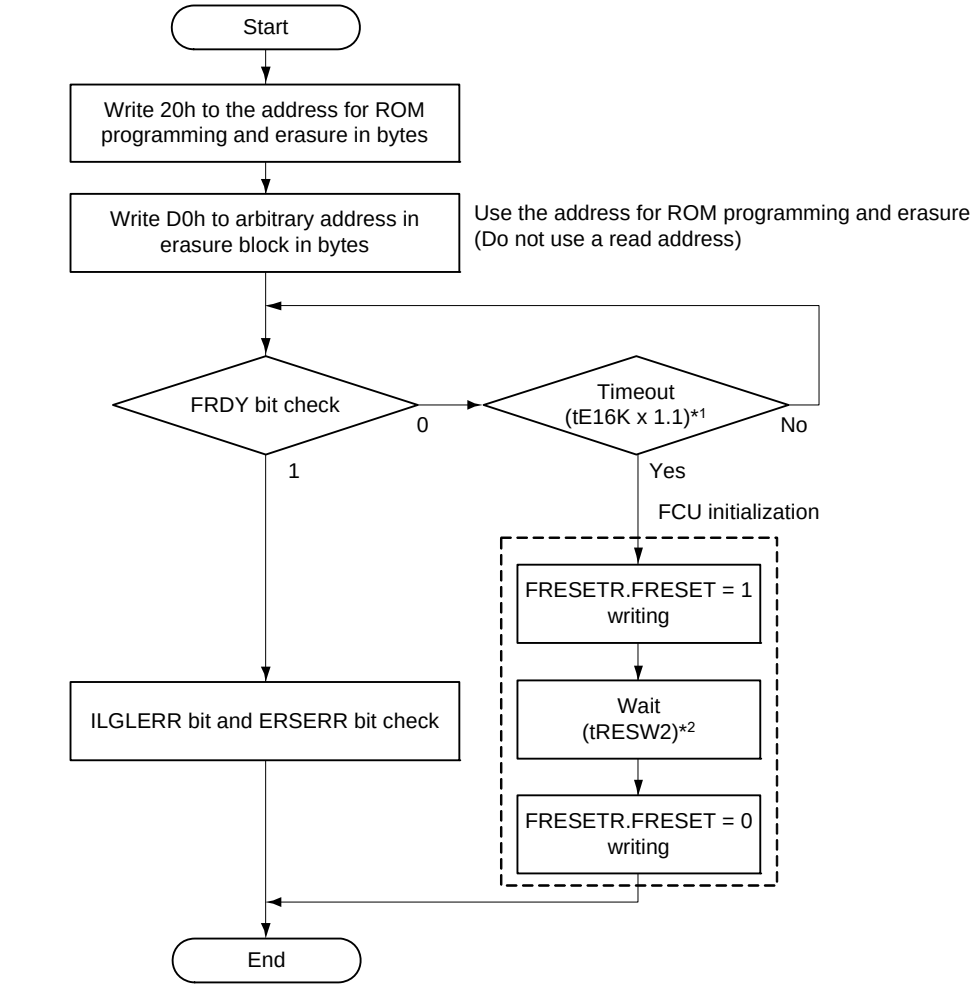


Note 1. tP128: Programming time for 128-byte data (see section 49, Electrical Characteristics)
Note 2. tRESW2: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

Figure 45.13 Procedure for ROM Programming

(6) Erasure

To erase data from the ROM, use the block erase command.
Write 20h to the ROM programming/erasure address in byte access in the first cycle of the block erase command. When D0h is written to an arbitrary address in a target erasure block in byte access in the second cycle, the FCU starts the erasure processing for the ROM. Whether erasure is completed can be checked with the FSTATR0.FRDY bit. Reading the erased ROM by the CPU returns FFFF FFFFh in 32 bits.
To execute an erasure with lock bit protection disabled, set the FPROTR.FPROTCN bit before erasure.

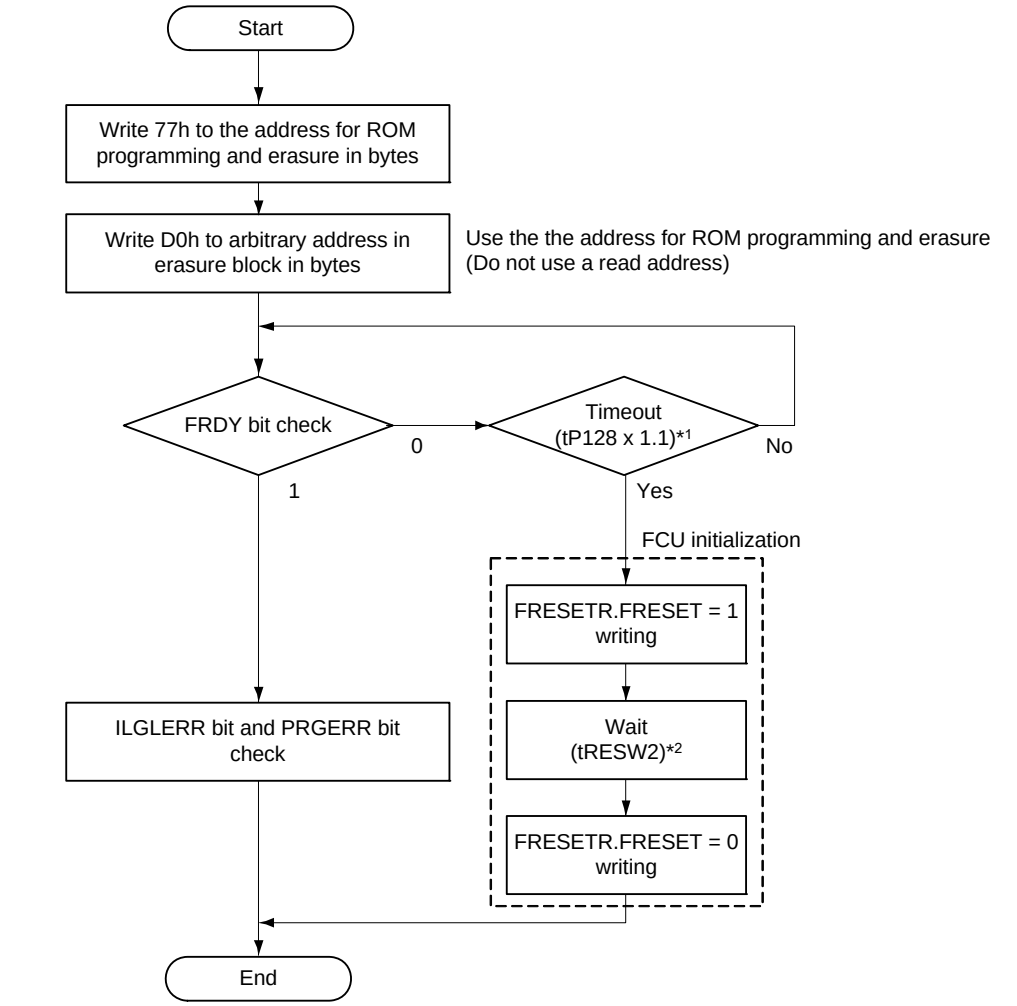


Note 1. tE16K: Erasure time for a 16-Kbyte erasure block (see section 49, Electrical Characteristics)
Note 2. tRESW2: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

Figure 45.14 Procedure for ROM Erasure

(7) Writing to/Erasing Lock Bits

Each erasure block in the user area includes a lock bit. To write to a lock bit, use the lock bit programming command. In the first cycle of the lock bit programming command, 77h is written to the ROM programming/erasure address as a byte. When D0h is written to an arbitrary address in an erasure block whose lock bit is to be written to in the second cycle as a byte, the FCU starts the processing to write to the lock bit. Whether writing is completed can be checked with the FSTATR0.FRDY bit.



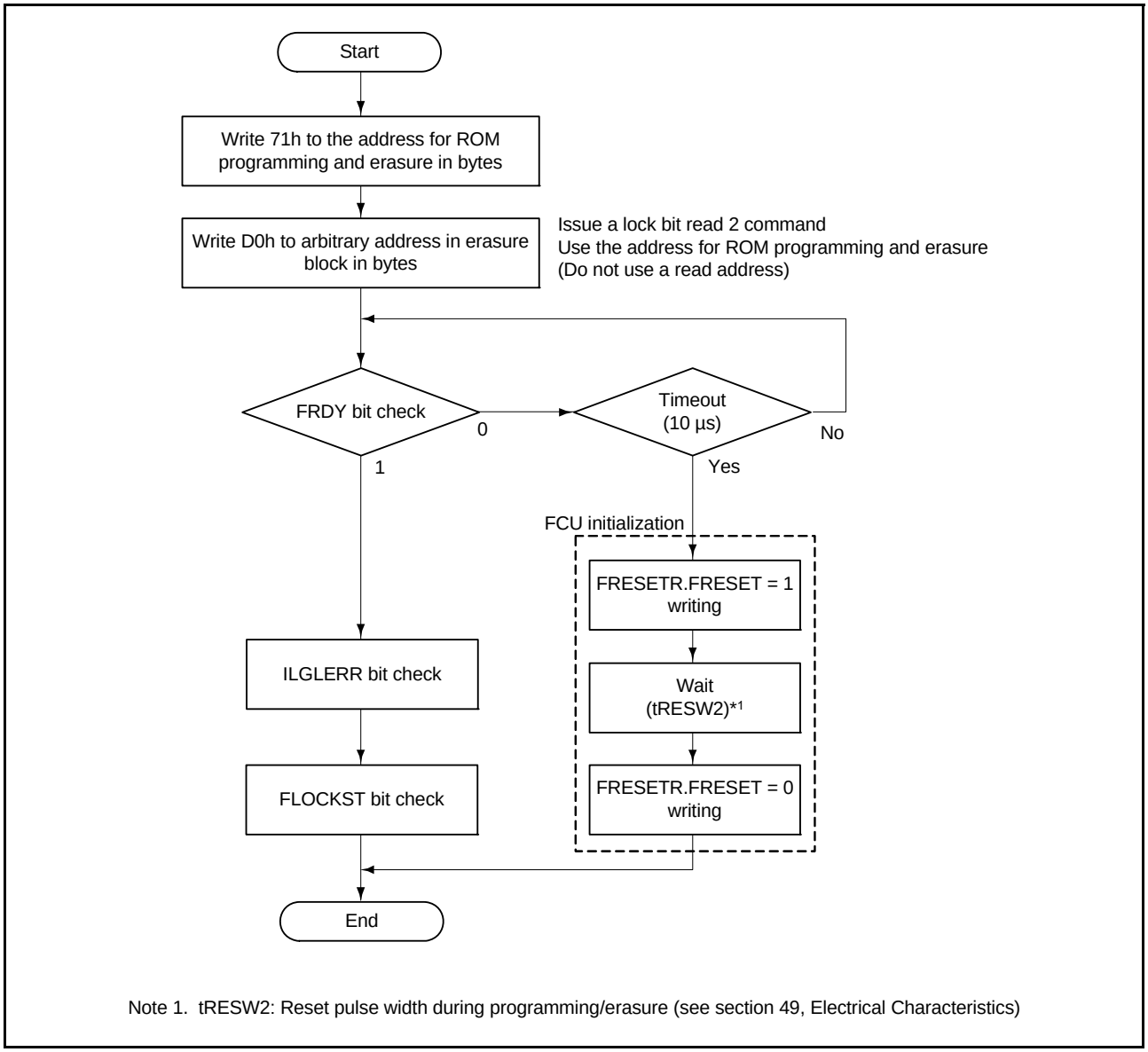
Note 1. tP128: Programming time for 128-byte data (see section 49, Electrical Characteristics)
Note 2. tRESW2: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

Figure 45.15 Procedure for Programming the Lock Bit

To erase a lock bit, use the block erase command.
When the FPROTR.FPROTCN bit is 0, erasure blocks whose lock bit is set to 0 cannot be erased. When erasing a lock bit, issue a block erase command with the FPROTCN bit set to 1. Using the block erase command erases all data in the erasure block. It is impossible to erase only a lock bit.

(8) Reading Lock Bits

Lock bits can be read by either reading from a memory area or reading from a register.
The lock bit read 2 command is issued in the case of the register reading method (i.e. when the FMODR.FRDMD bit is set to 1). This command is issued to an address within the erasure block for which the lock bit is to be read; the address range is that for programming and erasing the ROM. In the first and second cycles of the lock bit read 2 command, the values 71h and D0h are written as bytes; once these values have been written, the value of the lock bit for the specified erasure block is copied to the FSTATR1.FLOCKST bit.
In the case of the memory area reading method (i.e. when the FMODR.FRDMD bit is 0), the FCU is placed in ROM lock-bit read mode, and the lock bit is obtained by reading from an address within the address range for programming and erasure of the ROM. For details, see section 45.6.4.1, (5) Switching to ROM Lock-Bit Read Mode.



Note 1. tRESW2: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

Figure 45.16 Procedure for Reading Lock Bit in Register Read Mode

45.6.4.3 Error Processing

The following passages describe the flow of error processing. For details on errors, see section 45.8, Protection.

(1) Checking Flash Status Register 0 (FSTATR0)

To check FSTATR0, read FSTATR0 directly or read the ROM programming/erasure address in ROM status read mode. For the reading in ROM status read mode, see section 45.6.4.1, (4) Switching to ROM Status Read Mode.

(2) Clearing Flash Status Register 0 (FSTATR0)

To clear the ILGLERR, ERSERR and PRGERR bits in FSTATR0 to 0, use the status register clear command. When one of the ILGLERR, ERSERR and PRGERR bits in FSTATR0 is 1, the FCU is placed in the command-locked state and receives no FCU commands other than the status register clear command. If the ILGLERR bit is 1, also check the values of the ROMAE, DFLAE, DFLRPE, and DFLWPE bits in FASTAT. Even if issuing a status register clear command without clearing these bits, the ILGLERR bit is not cleared to 0.

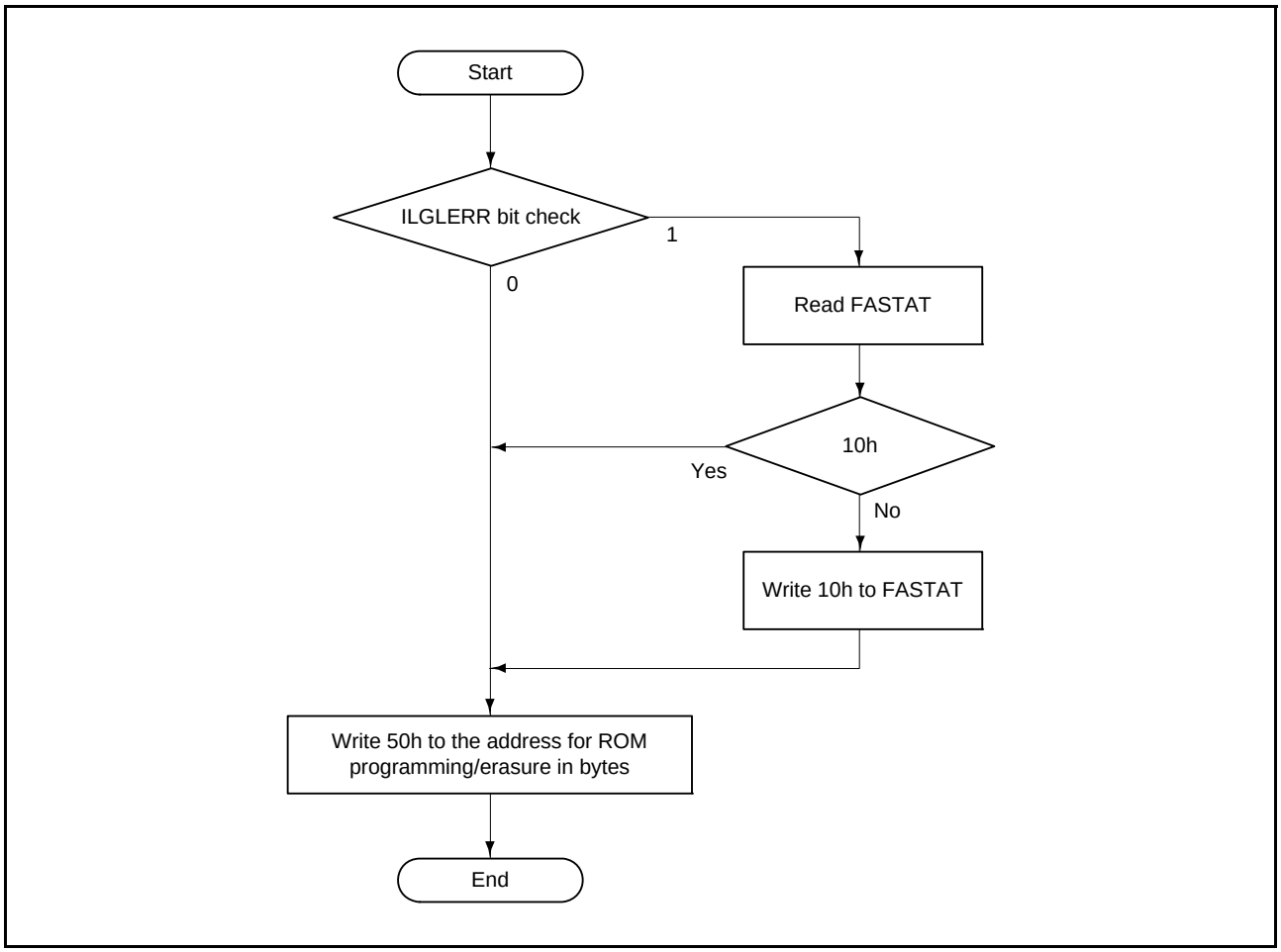


Figure 45.17 Procedure for Clearing FSTATR0

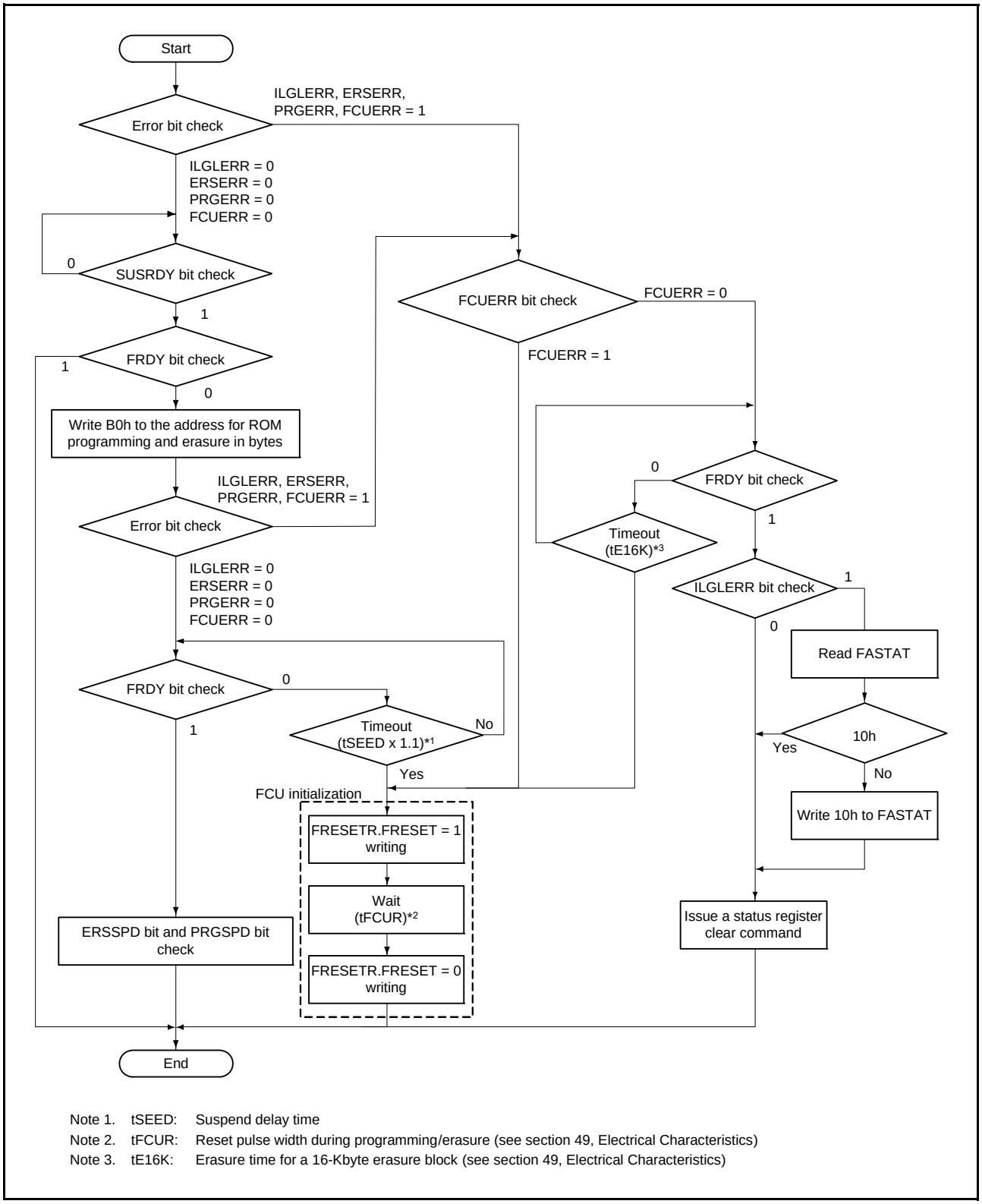
(3) Initializing the FCU

When a timeout leads to the FSTATR0.FRDY bit not being set to 1 after an FCU command has been issued, FRESETR must be used to initialize the FCU. FCU initialization by FRESETR is also necessary when the FSTATR1.FCUERR bit is 1. In either case, maintain the FRESETR.FRESET bit set to 1 for a period of tRESW2 (see section 49, Electrical Characteristics). Disable reading from the ROM and E2 DataFlash memory during this period of keeping the FRESET bit set to 1. In addition, while the FRESET bit is 1, FCU commands are disabled because FENTRYR is initialized. Restart the processing from the start, as shown in Figure 45.10.

45.6.4.4 Suspension and Resumption

(1) Suspending Programming or Erasure

To suspend programming/erasure for the ROM, use the P/E suspend command.
When issuing a P/E suspend command, check that the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FSTATR1.FCUERR bit are 0, and the execution of programming/erasure is normally performed. To confirm that the suspend command can be received, also check that the FSTATR0.SUSRDY bit is 1. After issuing a P/E suspend command, read FSTATR0 and FSTATR1 to confirm that no error occurs.
If an error occurs during programming/erasure, at least one of the ILGLERR, PRGERR, ERSERR, and FCUERR bits is set to 1. When programming/erasure processing has finished during the interval from when the SUSRDY bit is checked as 1 to when a P/E suspend command is received, the ILGLERR bit is set to 1 because the issued P/E suspend command is detected as an illegal command.
When programming/erasure processing has finished simultaneously with the reception of a P/E suspend command, no error occurs and the suspended state is not entered (FSTATR0.FRDY bit is 1 and ERSSPD and PRGSPD bits in FSTATR0 are 0). When a P/E suspend command is received and then the programming/erasure suspend processing finishes normally, the FCU enters the suspended state, the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is set to 1. After issuing a P/E suspend command, check that the ERSSPD or PRGSPD bit is 1 and the FCU enters the suspended state, and then decide the subsequent flow. When issuing a P/E resume command in the subsequent flow although the FCU does not enter the suspended state, an illegal command error occurs and the FCU is placed in the command-locked state (see section 45.8.2, Error Protection).
If the erasure suspended state is entered, programming to blocks other than an erasure target can be performed. Additionally, the programming and erasure suspended states can shift to ROM read mode by clearing FENTRYR.
For details on FCU operations at the reception of a P/E suspend command, see section 45.7, Suspending Operation.

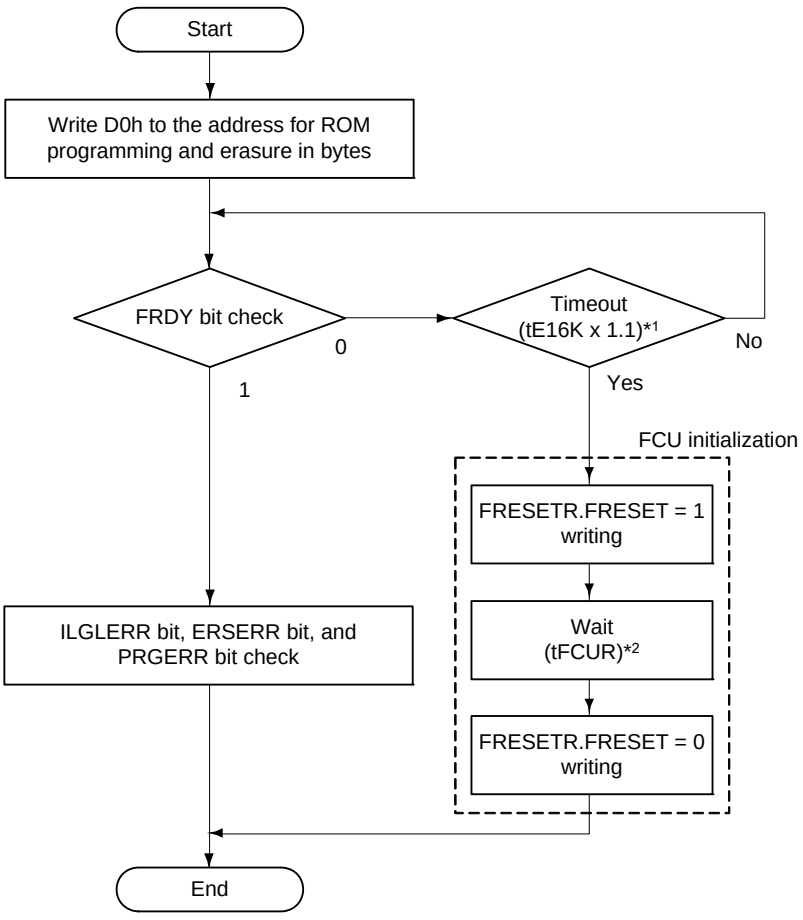


Note 1. tSEED: Suspend delay time
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)
Note 3. tE16K: Erasure time for a 16-Kbyte erasure block (see section 49, Electrical Characteristics)

Figure 45.18 Procedure for Programming/Erasure Suspension

(2) Resuming Programming or Erasure

To resume a suspended programming/erasure processing, use the P/E resume command. When the settings of FENTRYR are changed during suspension, reset FENTRYR to the value immediately before the P/E suspend command was issued, and then issue a P/E resume command.



Note 1. tE16K : Erasure time for a 16-Kbyte erasure block (see section 49, Electrical Characteristics)
Note 2. tFCUR : Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

Figure 45.19 Procedure for Resuming Programming or Erasure

45.7 Suspending Operation

The ROM cannot be read during programming/erasure. The ROM can be read by suspending the ROM programming/erasure with the P/E suspend command. The P/E suspend command includes two programming modes (suspension priority mode and erasure priority mode) and two erasure modes (suspension priority mode and erasure priority mode). The P/E resume command that resumes suspended programming/erasure processing is also provided.

45.7.1 Suspension during Programming

When issuing a P/E suspend command during the ROM programming, the FCU suspends programming processing. Figure 45.20 shows the suspend operation of programming.
When receiving a programming-related command, the FCU clears the FSTATR0.FRDY bit to 0 to start programming. If the FCU enters the state in which the P/E suspend command can be received after starting programming, the FSTATR0.SUSRDY bit is set to 1. When a P/E suspend command is issued, the FCU receives the command and clears the SUSRDY bit to 0. If the FCU receives a P/E suspend command while a programming pulse is being applied, the FCU continues applying the pulse. After the specified pulse application time, the FCU finishes pulse application, and starts the programming suspend processing and sets the FSTATR0.PRGSPD bit to 1. When the suspend processing finishes, the FCU sets the FRDY bit to 1 to enter the programming suspended state. When receiving a P/E resume command in the programming suspended state, the FCU clears the FRDY and PRGSPD bits to 0 and resumes programming.

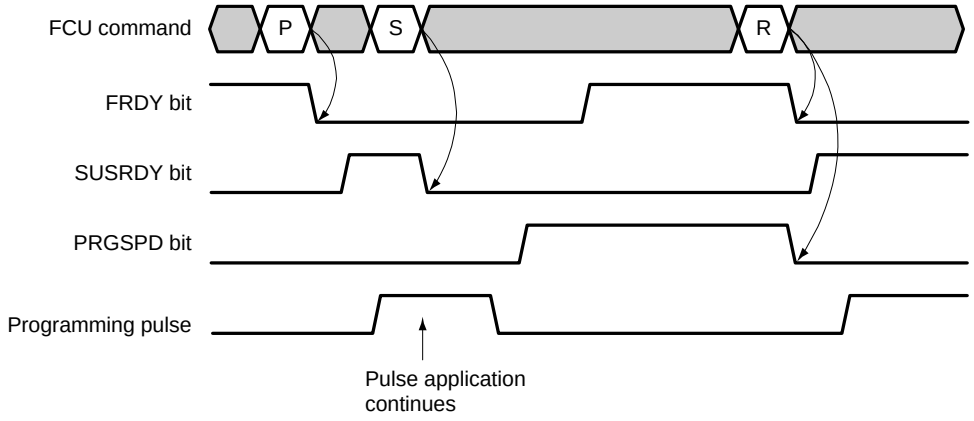


Figure 45.20 Suspension during Programming

45.7.2 Suspension during Programming/Erasure (Suspension Priority Mode)

The RX63N/RX631 has a suspension priority mode for the suspension of erasure.
Figure 45.21 shows the suspend operation of erasure when the erasure suspend mode is set to the suspension priority mode (FCPSR.ESUSPMD bit is 0).
When receiving an erasure-related command, the FCU clears the FSTATR0.FRDY bit to 0 to start erasure. If the FCU enters the state in which the P/E suspend command can be received after starting erasure, the FSTATR0.SUSRDY bit is set to 1. When a P/E suspend command is issued, the FCU receives the command and clears the SUSRDY bit to 0. When receiving a suspend command during erasure, the FCU starts the suspend processing and sets the FSTATR0.ERSSPD bit to 1 even if it is applying an erasure pulse. When the suspend processing finishes, the FCU sets the FRDY bit to 1 to enter the erasure suspended state. When receiving a P/E resume command in the erasure suspended state, the FCU clears the FRDY and ERSSPD bits to 0 and resumes erasure. Operations of the FRDY, SUSRDY, and ERSSPD bits at the suspension and resumption of erasure are the same, regardless of the erasure suspend mode.
The setting of the erasure suspend mode affects the control method of erasure pulses. In suspension priority mode, when receiving a P/E suspend command while erasure pulse A that has never been suspended in the past is being applied, the FCU suspends the application of erasure pulse A and enters the erasure suspended state. When receiving a P/E suspend command while reapplying erasure pulse A after erasure is resumed by a P/E resume command, the FCU continues applying erasure pulse A. After the specified pulse application time, the FCU finishes erasure pulse application and enters the erasure suspended state. When the FCU receives a P/E resume command next and erasure pulse B starts to be newly applied, and then the FCU receives a P/E suspend command again, the application of erasure pulse B is suspended. In suspension priority mode, delay due to suspension can be minimized because the application of an erasure pulse is suspended one time per pulse and priority is given to the suspend processing.

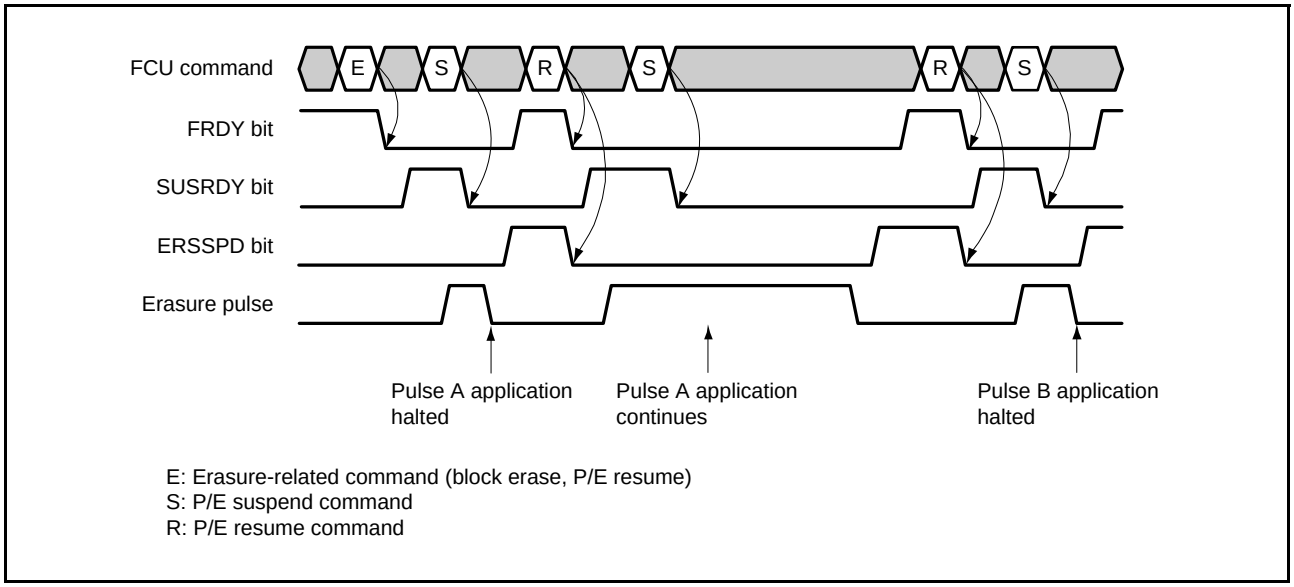


Figure 45.21 Suspension during Erasure (Suspension Priority Mode)

45.7.3 Suspension during Programming/Erasure (Programming/Erasure Priority Mode)

The RX63N/RX631 has a suspension priority mode for the suspension of erasure.
Figure 45.22 shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (FCPSR.ESUSPMD bit is 1). The control method of erasure pulses in erasure priority mode is the same as that of programming pulses for the programming suspend processing.
If the FCU receives a P/E suspend command while an erasure pulse is being applied, the FCU definitely continues applying the pulse. In this mode, the required time for the whole erasure processing can be reduced as compared with the suspension priority mode because the reapplication of erasure pulses does not occur when a P/E resume command is issued.

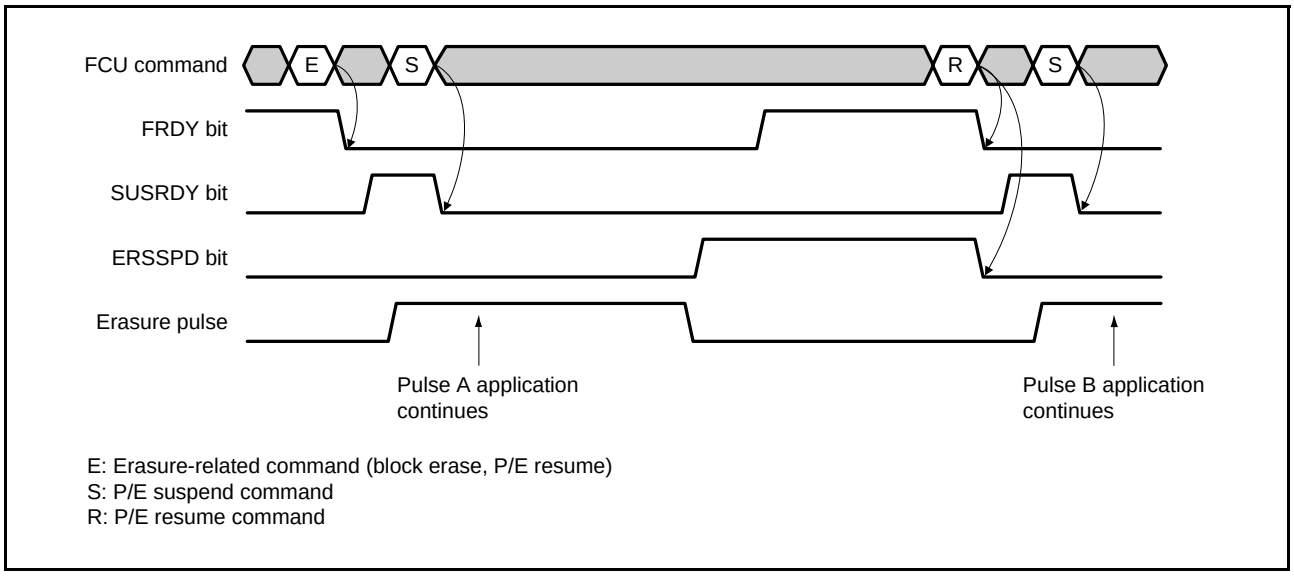


Figure 45.22 Suspension during Erasure (Erasure Priority Mode)

45.8 Protection

Protection against programming/erasure for the ROM includes software protection and error protection.

45.8.1 Software Protection

With the software protection, the ROM programming/erasure is prohibited by the settings of the control registers or user area lock bit. When the software protection is violated and a ROM programming/erasure-related command is issued, the FCU detects an error and enters the command-locked state.

(1) Protection through FWEPROR

If the FWEPROR.FLWE[1:0] bits are not set to 01b, programming cannot be performed in any of the modes.

(2) Protection through FENTRYR

When the FENTRYR.FENTRY3, FENTRY2, FENTRY1, FENTRY0 bits are 0, ROM read mode is selected. Because the FCU command cannot be received in ROM read mode, ROM programming/erasure is prohibited. When an FCU command is issued in ROM read mode, the FCU detects an illegal command error and is placed in the command-locked state (see section 45.8.2, Error Protection).

(3) Protection through Lock Bit

Each erasure block in the user area includes a lock bit. When the FPROTR.FPROTCN bit is 0, erasure blocks whose lock bit is set to 0 are prohibited from being programmed/erased. To program or erase erasure blocks whose lock bit is set to 0, set the FPROTCN bit to 1. When the lock bit protection is violated and a ROM programming/erasure-related command is issued, the FCU detects a programming/erasure error and enters the command-locked state (see section 45.8.2, Error Protection).

45.8.2 Error Protection

With the error protection, the FCU detects malfunctions caused by FCU command issuance errors and prohibited access occurrences, and an FCU command is prohibited from being received (command-locked state).
When the FCU enters the command-locked state (FASTAT.CMDLK bit is 1), one or several of the status bits (FSTATR0.ILGLERR, ERSERR, and PRGERR bits, FSTATR1.FCUERR bit, and FASTAT.ROMAE bit) are set to 1 and programming and erasure of the ROM are prohibited. To clear the command-locked state, a status register clear command must be issued with FASTAT set to 10h.
While the FAEINT.CMDLKIE bit is set to 1, if the FCU is placed in the command-locked state (FASTAT.CMDLK bit is set to 1), a flash interface error (FIFERR) interrupt occurs. While the FAEINT.ROMAEIE bit is set to 1, if the FASTAT.ROMAE bit is set to 1, an FIFERR interrupt occurs.
Table 45.9 lists the relationship between the contents of the ROM-related error protection and status bit values (FSTATR0.ILGLERR, ERSERR, and PRGERR bits, FSTATR1.FCUERR bit, and FASTAT.ROMAE bit) at error detection. If a command other than the suspend command is issued during programming/erasure and the FCU enters the command-locked state, it continues programming/erasure. In this state, it is impossible to issue a P/E suspend command and suspend programming/erasure. When a command is issued in the command-locked state, the ILGLERR bit is set to 1.

Table 45.9 Error Protection Types (Types Dedicated to ROM and Types Common to ROM and E^2 DataFlash)

Type	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
FENTRYR setting error	FENTRYR setting is other than AA01h, AA02h*1, AA04h*1, AA08h*1, and AA80h*1.	1	0	0	0	0
	The FENTRYR setting at suspension disagrees with that at resumption	1	0	0	0	0
Illegal command error	Undefined code is specified in the first cycle of an FCU command	1	0	0	0	0
	Other than D0h is specified in the last cycle of a multi-cycle FCU command	1	0	0	0	0
	The peripheral clock is set to other than 1 to 100 MHz in PCKAR (an error is not detected if the setting is from 1 to 4 MHz or from 50 to 100 MHz)	1	0	0	0	0
	A command other than the suspend command is issued during programming/erasure	1	0	0	0	0
	A suspend command is issued during processing other than programming/erasure	1	0	0	0	0
	A suspend command is issued in the suspended state	1	0	0	0	0
	A resume command is issued in other than the suspended state	1	0	0	0	0
	A programming/erasure-related command (programming/lock bit programming/block erase) is issued in the programming suspended state	1	0	0	0	0
	A block erase command is issued in the erasure suspended state	1	0	0	0	0
	A programming or lock bit programming command is issued to an erasure suspend target area in the erasure suspended state	1	0	0	0	0
	Other than 80h is specified in the second cycle of a programming command	1	0	0	0	0
	A command is issued in the command-locked state	1	0/1	0/1	0/1	0/1
Erasure error	An error occurs during erasure	0	1	0	0	0
	When the FPROTR.FPROTCN bit is 0, a block erase command is issued to an erasure block whose lock bit is set to 0	0	1	0	0	0
Programming error	An error occurs during programming	0	0	1	0	0
	When the FPROTR.FPROTCN bit is 0, a programming or lock bit programming command is issued to an erasure block whose lock bit is set to 0	0	0	1	0	0
FCU error	An error occurs during FCU internal processing	0	0	0	1	0
ROM access violation	When the FENTRYR.FENTRY0 bit is 1 in ROM P/E normal mode, a read access command is issued for addresses 00FA 0000h to 00FF FFFFh.	1	0	0	0	1
	When the FENTRYR.FENTRY1 bit is 1 in ROM P/E normal mode, a read access command is issued for addresses 00F4 0000h to 00F7 FFFFh when the user area capacity is 768 Kbytes or 00F0 0000h to 00F7 FFFFh when the user area capacity is 1 Mbyte or 1.5 Mbytes.*2	1	0	0	0	1
	When the FENTRYR.FENTRY2 bit is 1 in ROM P/E normal mode, a read access command is issued for addresses 00E8 0000h to 00EF FFFFh.*2	1	0	0	0	1
	When the FENTRYR.FENTRY3 bit is 1 in ROM P/E normal mode, a read access command is issued for addresses 00E0 0000h to 00E7 FFFFh.*2	1	0	0	0	1
	When the FENTRYR.FENTRY0 bit is 0, a read access command is issued for addresses 00FA 0000h to 00FF FFFFh.	1	0	0	0	1
	When the FENTRYR.FENTRY0 bit is 0, a read access command is issued for addresses 00F4 0000h to 00F7 FFFFh when the user area capacity is 768 Kbytes or 00F0 0000h to 00F7 FFFFh when the user area capacity is 1 Mbyte or 1.5 Mbytes.*2	1	0	0	0	1
	When the FENTRYR.FENTRY2 bit is 0, a read access command is issued for addresses 00E8 0000h to 00EF FFFFh.*2	1	0	0	0	1
	When the FENTRYR.FENTRY3 bit is 0, a read access command is issued for addresses 00E0 0000h to 00E7 FFFFh.*2	1	0	0	0	1
	A read access command is issued for addresses FFF4 0000h/FFF0 0000h/FFE8 0000h*3 to FFFF FFFFh while FENTRYR setting is other than 0000h.	1	0	0	0	1

Note 1. AA02h, AA04h, and AA08h can be written to FENTRYR when the user area capacity is 768 Kbytes, 1 Mbyte, 1.5 Mbytes, or 2 Mbytes, 1.5 Mbytes or 2 Mbytes, and 2 Mbytes, respectively.

Note 2. The FENTRY1, FENTRY2, and FENTRY3 bits are provided when the user area capacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively.

Note 3. FFFE 0000h, FFE8 0000h, FFF0 0000h, and FFF4 0000h are applicable when the user area capacity is 2 Mbytes, 1.5 Mbytes, 1 Mbyte, 768 Kbytes, respectively.

45.9 User Boot Mode

When user boot mode is set by the MD0 pin and reset is canceled, a transition to user boot mode is made. The reset vector at this time points to the address FF7F FFFCh of the user boot area. The other vector tables refer to normal vector table (see section 15, Interrupt Controller (ICUb)).
In user boot mode, it is possible to perform programming using a given interface; user area or data can be programmed or erased by issuing the FCU command. Note that programming to the user boot area should be performed in boot mode.

45.10 Boot Mode

45.10.1 System Configuration

In boot mode, the host sends control commands and data for programming, and the user area and data area are programmed or erased accordingly. An on-chip SCI handles transfer between the host and RX63N/RX631 in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.
When the RX63N/RX631 is activated in boot mode, the program on the area that holds the embedded program is executed. This program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.
Figure 45.23 shows the system configuration for operations in boot mode.

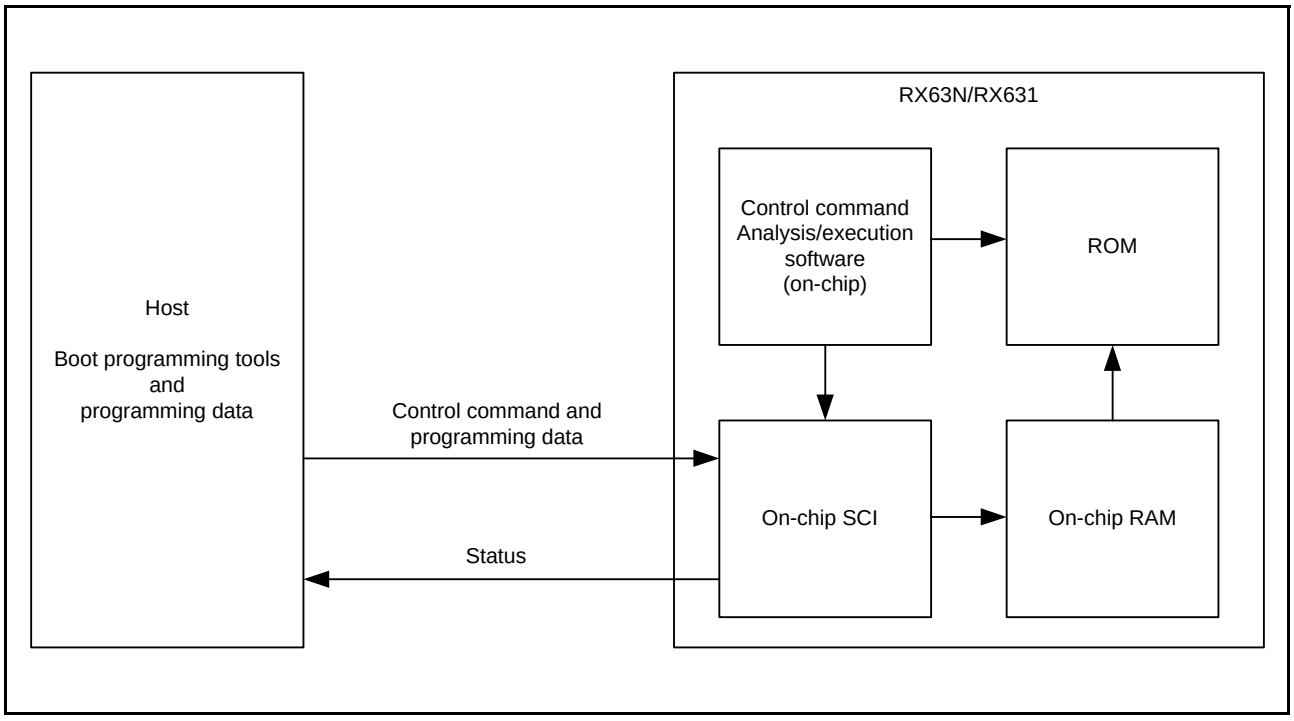


Figure 45.23 System Configuration for Operations in Boot Mode

45.10.2 State Transitions in Boot Mode

Figure 45.24 is a diagram of the state transitions in boot mode.

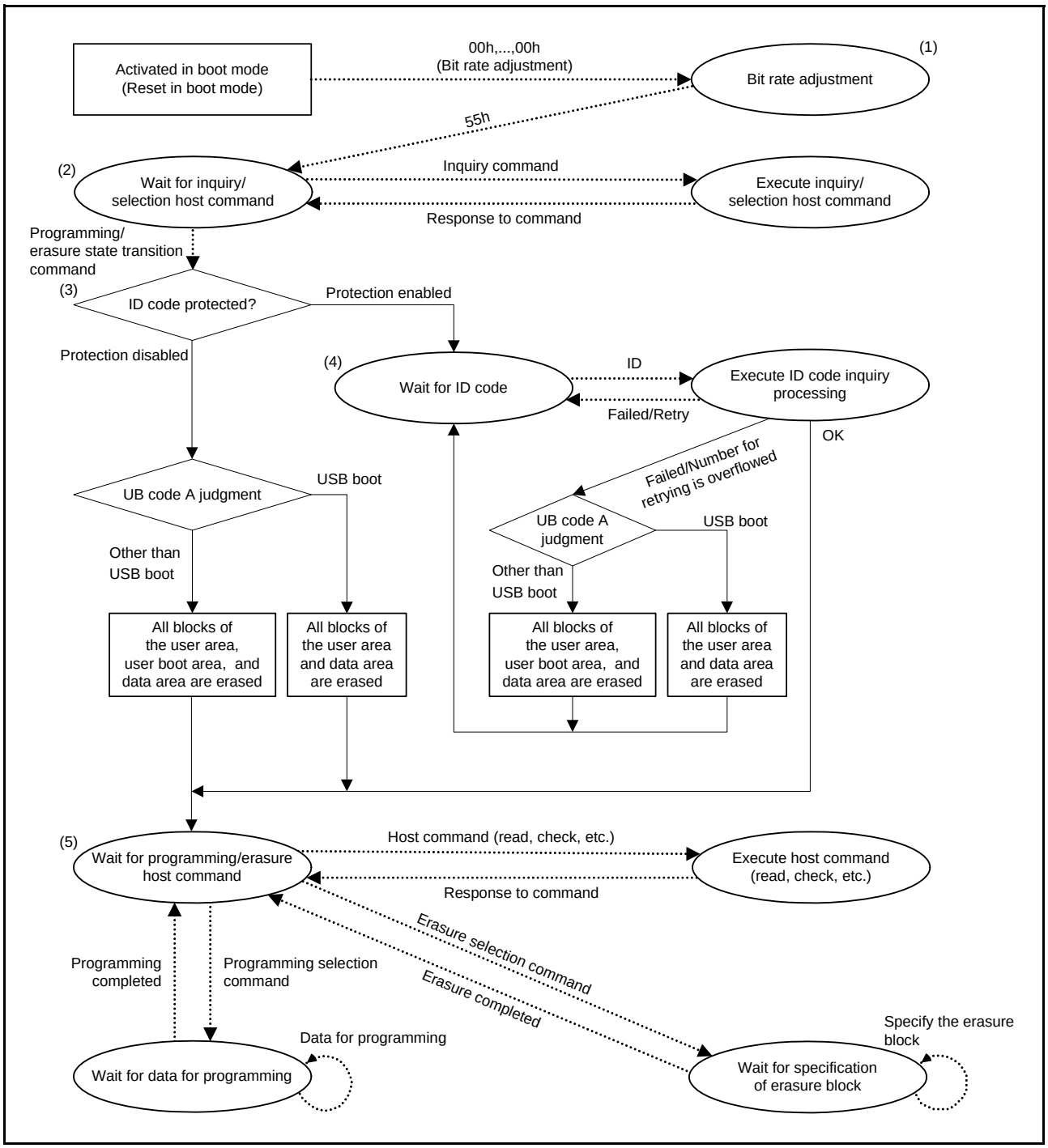


Figure 45.24 State Transitions in Boot Mode

(1) Matching the Bit Rates

When the RX63N/RX631 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, the RX63N/RX631 transmits the value 00h to the host. On subsequent correct reception of the value 55h sent from the host, the RX63N/RX631 enters the state of waiting for a host command for inquiry or selection. For details on matching of the bit rates, see section 45.10.3, Automatic Adjustment of the Bit Rate.

(2) Waiting for a Host Command for Inquiry or Selection

This state is for inquiries on the area size, the area configuration, the addresses where the areas start, the state of support etc., and for selection of the device, clock mode, and bit rate. The RX63N/RX631 receives a programming/erasure state transition command issued by the host and then enters the state to determine whether ID code protection is enabled or disabled. For the inquiry/selection host commands, see section 45.10.6, Inquiry/Selection Host Command Wait State.

(3) Judging ID Code Protection

This state is for determining whether ID code protection is enabled or disabled. The control code and ID code written in ROM are used to determine whether ID code protection is enabled or disabled. When enabled, the state of waiting for the ID code is entered. When disabled, the user area and data area are completely erased, and the state of waiting for programming and erasure commands from the host is entered. For details on the control code and ID code, see section 45.10.4, ID Code Protection (Boot Mode).

(4) Waiting for an ID Code

This state is for waiting for the control code and ID code to be sent from the host. The control code and ID code sent by the host are compared with the code stored in ROM, and the state of waiting for programming and erasure commands from the host is entered if the two match. If they do not match, the next transition is back to the state of waiting for an ID code. However, if the ID codes fail to match three times in a row and also the state of protection is authentication method 1, the ROM is completely erased, and the state of waiting for an ID code is entered again. A reset is required to release the system from this state due to non-matching ID codes. For details on the control code and ID code, see section 45.10.4, ID Code Protection (Boot Mode).

(5) Waiting for a Host Command for Programming or Erasure

In this state, programming and erasure proceed in accordance with commands from the host. In response to the reception of a command, the RX63N/RX631 enters the state of waiting for the data to use in programming, the state of waiting for specification of the erasure block to be erased, or the state of executing the processing of commands, such as read and check.

When the RX63N/RX631 receives a programming selection command, it enters the state of waiting for the data to use in programming. After the host has issued the programming selection command, the process continues with the address where programming is to start and then the data for programming. Setting of FFFF FFFFh as the address where programming is to start indicates the completion of programming, and the next transition is from the state of waiting for the data to use in programming to the state of waiting for programming and erasure commands.

When the RX63N/RX631 receives an erasure selection command, it enters the state of waiting for specification of the erasure block to be erased. After the host has issued the erasure selection command, the process continues with the number of the erasure block to be erased. Setting of FFh as the number of the erasure block indicates the completion of erasure, and the next transition is from the state of waiting for specification of the erasure block to the state of waiting for programming and erasure commands. Since the user area, user boot area and data area are all completely erased during the interval between booting up in boot mode and transition to the state of waiting for programming and erasure commands, execution of erasure is not necessary unless data newly programmed in boot mode is to be erased without a further reset.

Other than the programming and erasure commands, commands from the host for execution in this state include those for sum checking of the user area and user boot area, blank checking (to confirm erasure), reading from memory, and

acquiring status information.

45.10.3 Automatic Adjustment of the Bit Rate

When the RX63N/RX631 is booted up in boot mode, 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 or 19,200 bps. The RX63N/RX631 calculates the host's SCI bit rate from the measured periods at low level, adjusts its own bit rate accordingly, 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 the RX63N/RX631. If successful reception of 00h by the host is not possible, reboot the RX63N/RX631 in boot mode, and then repeat the process of automatically adjusting the bit rate. If reception of the value 55h by the RX63N/RX631 is successful, it responds by sending E6h to the host, and if successful reception of 55h by the RX63N/RX631 is not possible, it responds by sending FFh to the host.

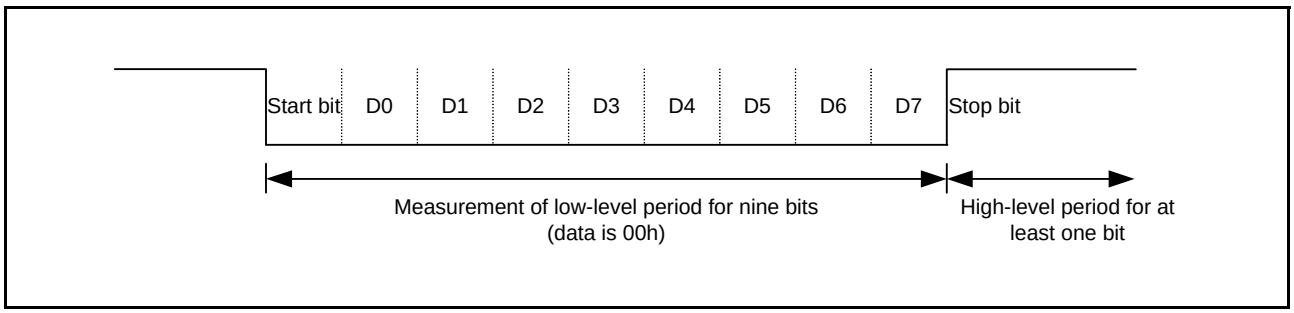


Figure 45.25 Transfer Format Used by SCI in Automatic Adjustment of Bit Rate

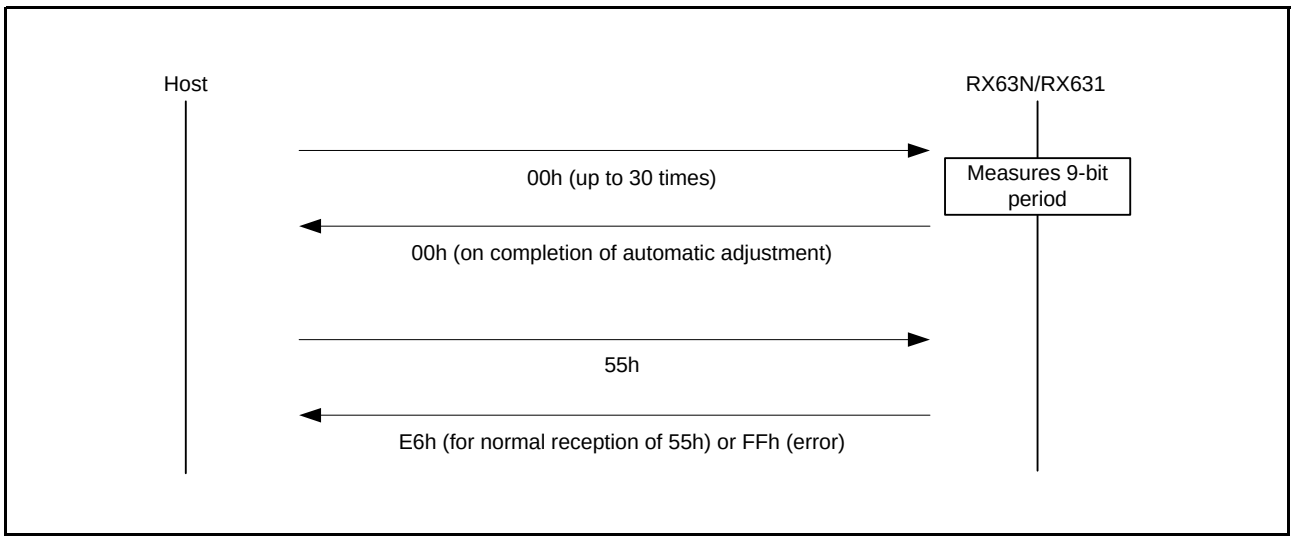


Figure 45.26 Sequence of Transfer between Host and RX63N/RX631

Since the bit rate of the RX63N/RX631 depends on the bit rate of the host's SCI module and the frequency of the RX63N/RX631's peripheral clock, adjustment to match the bit rate will not be possible under some conditions. Accordingly, ensure that SCI communication is under the conditions given in Table 45.10.

Table 45.10 Conditions for Automatic Bit-Rate Adjustment

Bit Rate of SCI in Host	Frequency Range for EXTAL Signal
9,600 bps	4 to 20 MHz*1
19,200 bps	8 to 20 MHz*1

Note 1. The maximum frequency of the resonator is 16 MHz.

45.10.4 ID Code Protection (Boot Mode)

This function is used to prohibit reading/programming/erasure from the host such as the PC.
After automatic adjustment of the bit rate when booting up in boot mode, the ID code transmitted from the host and the control and ID codes written to the ROM are used to determine disabling or enabling of ID code protection. When ID code protection is enabled, the code sent from the host is compared with the control code and ID code in the ROM to determine whether they match, and reading/programming/erasure will be enabled only when the two match.
The control code and ID code in the ROM consists of four 32-bit words. Figure 45.27 shows the configuration of the control code and ID code. The ID code should be set in 32-bit units.

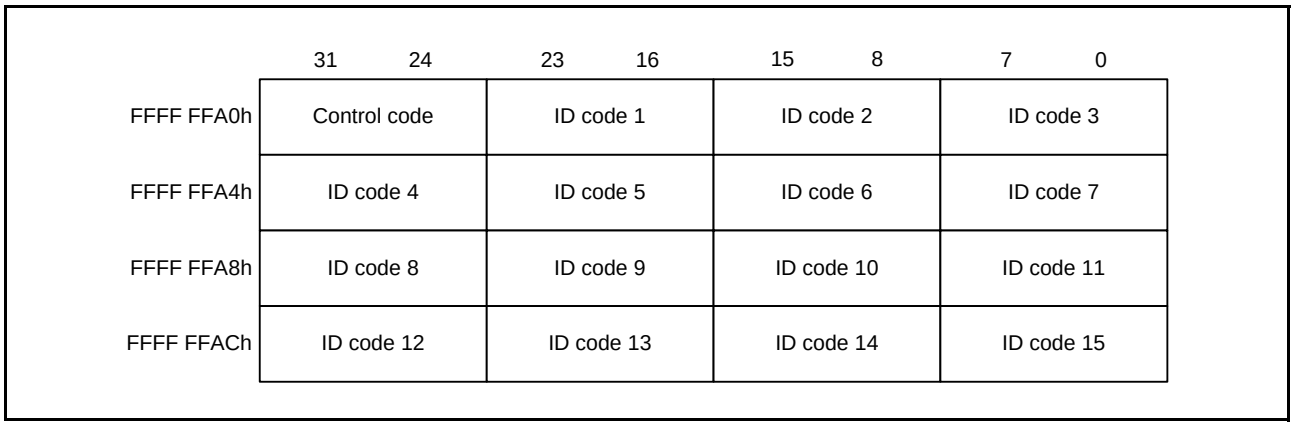


Figure 45.27 Configuration of Control Code and ID Code in ROM

(1) Control Code

The control code determines whether ID code protection is enabled or disabled and the method of authentication to use with the host. Table 45.11 lists how the control code determines the method of authentication

Table 45.11 Specifications for ID Code Protection

Control Code	ID Code	State of Protection	Operations at the Time of SCI Connection
45h	As desired	Protection enabled (authentication method 1)	Matching ID code: ID code protection is canceled and the host command waiting state is entered. Non-matching ID code: The ID code protection waiting state is entered again. However, if a non-matching ID code is received three times in a row, all blocks are erased.
52h	Sequences other than 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled (authentication method 2)	Matching ID code: ID code protection is canceled and the host command waiting state is entered. Non-matching ID code: The ID code protection waiting state is entered again.
	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled (authentication method 3)	Always judged to be a non-matching ID code.
Other than above	—	Protection disabled	All blocks are erased.

(2) ID Code

The ID code can be set to any desired value. However, if the control code is 52h and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh (from the ID code 1 field), there is no determination of matching and the ID code is always considered to be non-matching. Accordingly, reading, programming, and erasure from the host are prohibited.

(3) Program Example for ID Code Setting

The following assembler directives set up a control code of 45h and an ID code of 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh (from the ID code 1 field).

```
.SECTION  ID_CODE,CODE
.ORG  0FFFFFFA0h
.LWORD  45010203h
.LWORD  04050607h
.LWORD  08090A0Bh
.LWORD  0C0D0E0Fh
```

45.10.5 UB Code

For the UB code, see section 7.3.1, UB Code A.

45.10.6 Inquiry/Selection Host Command Wait State

Table 45.12 lists the host commands available in the inquiry/selection host command wait state. The embedded program status inquiry command can also be used in the programming/erasure host command wait state. The other commands can only be used in the inquiry/selection host command wait state.

Table 45.12 Inquiry/Selection Host Commands

Host Command Name	Function
Supported device inquiry	Inquires regarding the device codes and the product codes for the embedded programs
Device selection	Selects a device code
Clock mode inquiry	Inquires regarding the number of clock modes and their values
Clock mode selection	Notifies the selected clock mode
Multiplication ratio inquiry	Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication /division ratios
Operating frequency inquiry	Inquires regarding the number of clock types and the maximum and minimum operating frequencies
User boot area information inquiry	Inquires regarding the number of user boot area and the start and end addresses
User area information inquiry	Inquires regarding the number of user area and the start and end addresses
Erasure block information inquiry	Inquires regarding the number of blocks and the start and end addresses
Programming size inquiry	Inquires regarding the size of programming data
New bit rate selection	Modifies the bit rate of SCI communications between the host and RX63N/RX631
Programming/erasure state transition	Enters the state for determining ID code protection
Embedded program status inquiry	Inquires regarding the processing state

If the host has sent an undefined command, the RX63N/RX631 returns a response indicating a command error in the format shown below. The command field holds the first byte of the undefined command sent from the host.

Error response	80h	Command

In the inquiry/selection host command wait state, send selection commands from the host in the order of device selection, clock mode selection, and new bit rate selection to set up the RX63N/RX631 according to the responses to inquiry commands. Note that the supported device inquiry and clock mode inquiry commands are the only inquiry commands that can be sent before the clock mode selection command; other inquiry commands must not be issued before the clock mode selection command. If commands are issued in an incorrect order, the RX63N/RX631 returns a response indicating a command error. Figure 45.28 shows an example of the procedure to use host commands in the inquiry/selection host command wait state.

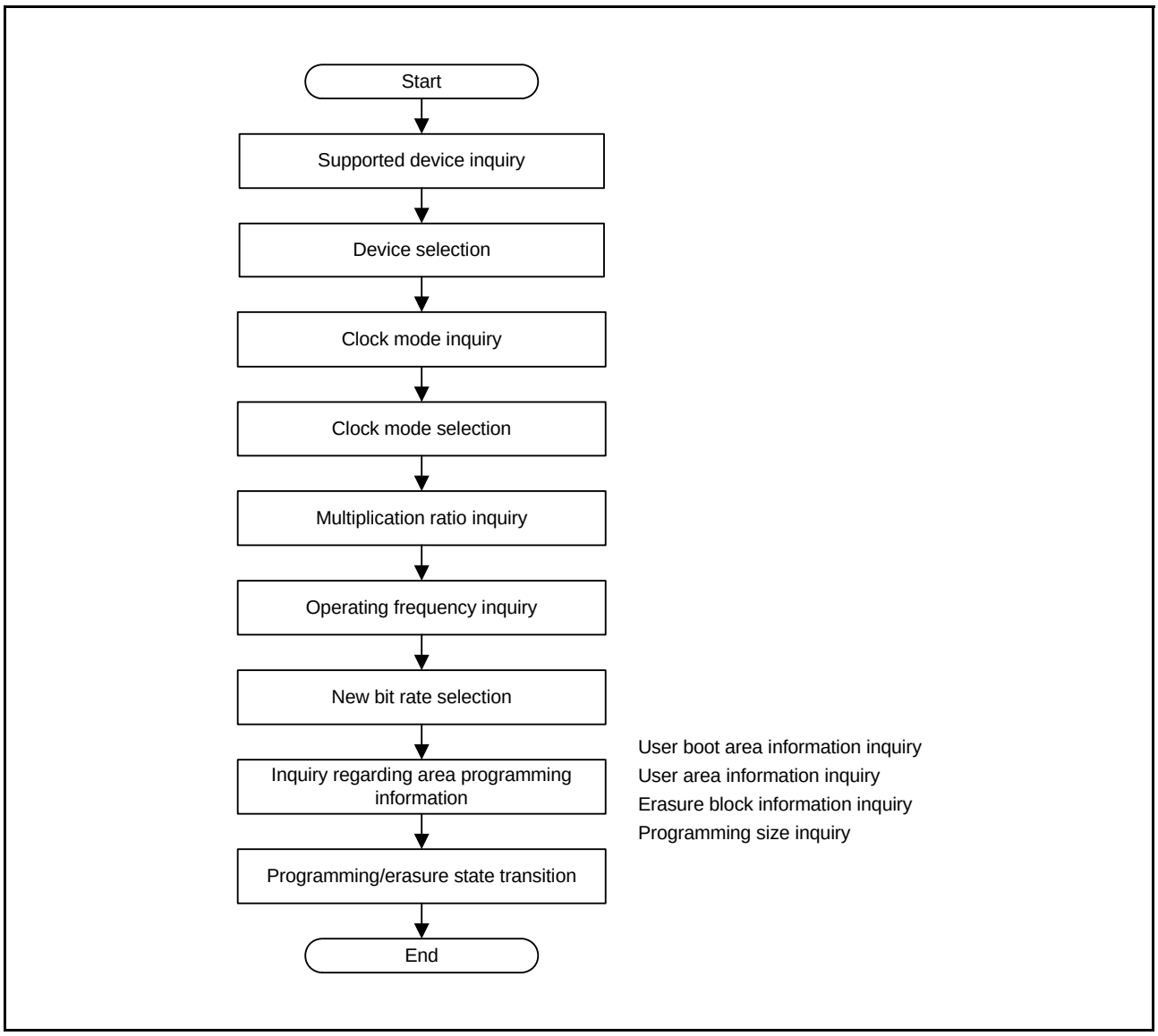


Figure 45.28 Example of Procedure to Use Inquiry/Selection Host Commands for User Area/User Boot Area

Each host command is described in detail below. The “command” in the description indicates a command sent from the host to the RX63N/RX631 and the “response” indicates a response sent from the RX63N/RX631 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX63N/RX631 becomes 00h.

(1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, the RX63N/RX631 returns the information concerning the devices supported by the embedded program for boot mode. If the supported device inquiry command comes after the host has selected a device, the RX63N/RX631 only returns the information concerning the selected device. In response to supported device inquiry commands, the RX63N/RX631 transmits two sets of device information in turn, one in little endian and the other in big endian.

Command	20h			
Response	30h	Size	Device count	
	Character count	Device code (little endian is specified)		Product code
	Character count	Device code (big endian is specified)		Product code
	SUM			

Size (1 byte):	Total number of bytes in the device count, character count, device code, and product code fields
Device count (1 byte):	Number of device types supported by the embedded program for boot mode
Character count (1 byte):	Number of characters included in the device code and product code fields
Device code (4 bytes):	ASCII code for the product name of the chip
Product code (n bytes):	ASCII code for the supported device
SUM (1 byte):	Checksum

(2) Device Selection

In response to a device selection command sent from the host, the RX63N/RX631 checks if the selected device is supported. When the selected device is supported, the RX63N/RX631 specifies this device as the device for use and returns a response (06h). If the selected device is not supported or the sent command is illegal, the RX63N/RX631 returns an error response (90h).
Select the device code with the endian specification from the two sets of device information transmitted in response to a supported device inquiry command in accord with the written data.

Command	10h	Size	Device code	SUM
Response	06h			
Error response	90h	Error		

Size (1 byte):	Number of characters in the device code field (fixed at 4)
Device code (4 bytes):	ASCII code for the product name of the chip (same code as the response to the supported device inquiry command)
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error (illegal command) 21h: Incorrect device code error

(3) Clock Mode Inquiry

In response to a clock mode inquiry command sent from the host, the RX63N/RX631 returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, the RX63N/RX631 only returns the information concerning the selected clock mode.

Command	21h			
Response	31h	Size		
	Mode	Mode	. . .	Mode
	SUM			

Size (1 byte): Total number of bytes in the mode count and mode fields
Mode (1 byte): Supported clock mode (for example, 01h indicates clock mode 1)
SUM (1 byte): Checksum

(4) Clock Mode Selection

In response to a clock mode selection command sent from the host, the RX63N/RX631 checks if the selected clock mode is supported. When the selected mode is supported, the RX63N/RX631 specifies this clock mode for use and returns a response (06h). If the selected mode is not supported or the sent command is illegal, the RX63N/RX631 returns an error response (91h).

Be sure to issue a clock mode selection command only after issuing a device selection command. Even when 00h or 01h has been returned as the number of supported clock modes in response to a clock mode inquiry command, issue a clock mode selection command to specify the clock mode that has been returned as the result of the inquiry.

Command	11h	Size	Mode	SUM
Response	06h			
Error response	91h	Error		

Size (1 byte): Number of characters in the mode field (fixed at 1)
Mode (1 byte): Clock mode (same mode as the response to the clock mode inquiry command)
SUM (1 byte): Checksum
Error (1 byte): Error code
11h: Checksum error (illegal command)
21h: Incorrect clock mode error

(5) Multiplication Ratio Inquiry

In response to a multiplication ratio inquiry command sent from the host, the RX63N/RX631 returns the clock types, the number of multiplication/division ratios, and the multiplication division ratios supported.

Command

22h

Response

32h	Size	Clock type count		
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
. . .	. . .	. . .	. . .	. . .
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
SUM				

Size (1 byte): Total number of bytes in the clock type count, multiplication ratio count, and multiplication ratio fields

Clock type count (1 byte): Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)

Multiplication ratio count (1 byte): Number of supported multiplication/division ratios (for example, 04h indicates that four multiplication ratios are supported for the system clock (multiplied by 1, multiplied by 2, multiplied by 4, and multiplied by 8))

Multiplication ratio (1 byte): A positive value indicates a multiplication ratio (for example, 04h = 4 = multiplied by 4)
A negative value indicates a division ratio (for example, FEh = –2 = divided by 2)

SUM (1 byte): Checksum

(6) Operating Frequency Inquiry

In response to an operating frequency inquiry command sent from the host, the RX63N/RX631 returns the minimum and maximum operating frequencies for each clock.

Command

23h

Response

33h	Size	Clock type count
Minimum frequency		Maximum frequency
Minimum frequency		Maximum frequency
. . .		. . .
Minimum frequency		Maximum frequency
SUM		

Size (1 byte): Total number of bytes in the clock type count, minimum frequency, and maximum frequency fields

Clock type count (1 byte): Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)

Minimum frequency (2 bytes): Minimum value of the operating frequency (for example, 07D0h indicates 20.00 MHz).
This value should be calculated by multiplying the frequency value (MHz) to two decimal places by 100.

Maximum frequency (2 bytes): Maximum value of the operating frequency
This value is represented in the same format as the minimum frequency

SUM (1 byte): Checksum

(7) User Boot Area Information Inquiry

In response to a user boot area information inquiry command sent from the host, the RX63N/RX631 returns the number of user boot areas and their addresses.

Command	24h

Response	34h	Size	Area count
	Area start address		
	Area end address		
	Area start address		
	Area end address		
	. . .		
	Area start address		
	Area end address		
	SUM		

Size (1 byte):	Total number of bytes in the area count, area start address, and area end address fields
Area count (1 byte):	Number of user boot areas (consecutive areas are counted as one area)
Area start address (4 bytes):	Start address of a user boot area
Area end address (4 bytes):	End address of a user boot area
SUM (1 byte):	Checksum

(8) User Area Information Inquiry

In response to a user area information inquiry command sent from the host, the RX63N/RX631 returns the number of user areas and their addresses.

Command	25h

Response	35h	Size	Area count
	Area start address		
	Area end address		
	Area start address		
	Area end address		
	. . .		
	Area start address		
	Area end address		
	SUM		

Size (1 byte):	Total number of bytes in the area count, area start address, and area end address fields
Area count (1 byte):	Number of user areas (consecutive areas are counted as one area)
Area start address (4 bytes):	Start address of a user area
Area end address (4 bytes):	End address of a user area
SUM (1 byte):	Checksum

(9) Erasure Block Information Inquiry

In response to an erasure block information inquiry command sent from the host, the RX63N/RX631 returns the number of total erasure blocks in the user area and data area, and their addresses.

Command	26h		
Response	36h	Size	Block count
	Block start address		
	Block end address		
	Block start address		
	Block end address		
	. . .		
	Block start address		
	Block end address		
	SUM		

Size (2 bytes):	Total number of bytes in the block count, block start address, and block end address fields
Block count (1 byte):	Number of erasure blocks in the user area
Block start address (4 bytes):	Start address of an erasure block
Block end address (4 bytes):	End address of an erasure block
SUM (1 byte):	Checksum

(10) Programming Size Inquiry

In response to a programming size inquiry command sent from the host, the RX63N/RX631 returns the programming size.

Command	27h			
Response	37h	Size	Programming size	SUM

Size (1 byte):	Number of characters included in the programming size field (fixed at 2)
Programming size (2 bytes):	Programming unit (bytes)
SUM (1 byte):	Checksum

(11) New Bit Rate Selection

In response to a new bit rate selection command sent from the host, the RX63N/RX631 checks if the on-chip SCI can be set to the selected new bit rate. When the SCI can be set to the new bit rate, the RX63N/RX631 returns a response (06h) and sets the SCI to the new bit rate. If the SCI cannot be set to the new bit rate or the sent command is illegal, the RX63N/RX631 returns an error response (BFh). Upon reception of response 06h, the host waits for a one-bit period in the previous bit rate with which the new bit rate selection command has been sent, and then sets the host's bit rate to the new one. After that, the host sends confirmation data (06h) in the new bit rate, and the RX63N/RX631 returns a response (06h) for the confirmation data.

Be sure to issue a new bit rate selection command only after a clock mode selection command.

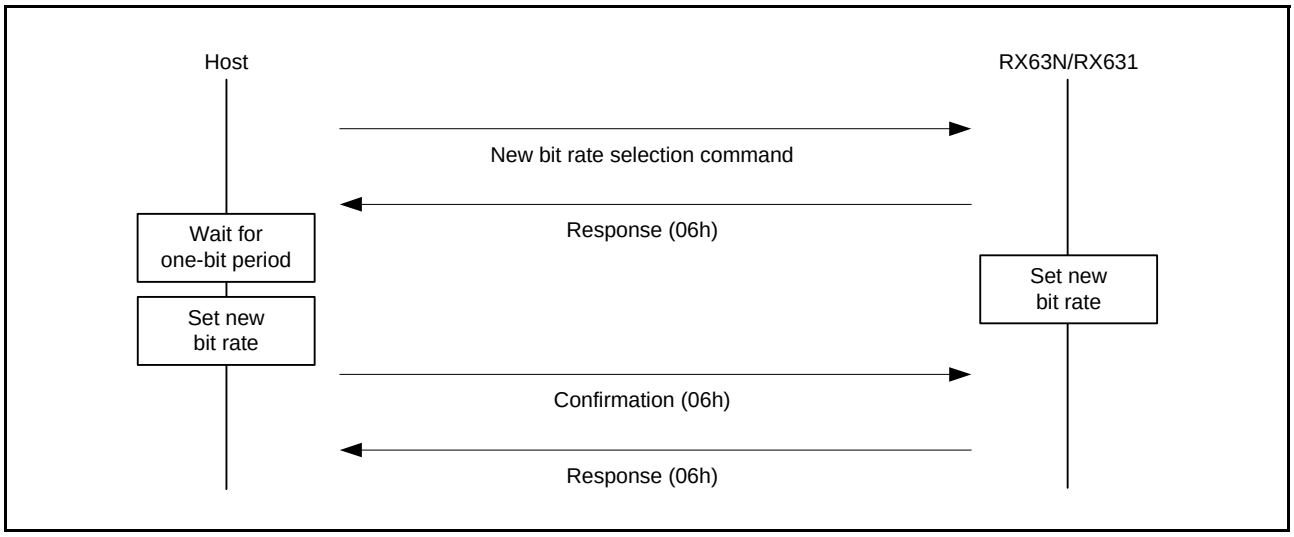


Figure 45.29 New Bit Rate Selection Sequence

Command	3Fh	Size	Bit rate	Input frequency
	Clock type count	Multiplication ratio 1	Multiplication ratio 2	
	SUM			
Response	06h			
Error response	BFh	Error		
Confirmation	06h			
Response	06h			

Size (1 byte):	Total number of bytes in the bit rate, input frequency, clock type count, and multiplication ratio fields
Bit rate (2 bytes):	New bit rate (for example, 00C0h indicates 19200 bps) 1/100 of the new bit rate value should be specified.
Input frequency (2 bytes):	Frequency input to the RX63N/RX631 (for example, 04E2h indicates 12.50 MHz) This value should be calculated by multiplying the input frequency value to two decimal places by 100.
Clock type count (1 byte):	Number of clock types (fixed: 02h indicates two clock types; that is, a system clock and a peripheral clock)
Multiplication ratio 1 (1 byte):	Multiplication/division ratio of the input frequency to obtain the system clock (ICLK) A positive value indicates a multiplication ratio (for example, 04h = 4 = multiplied by 4) A negative value indicates a division ratio (for example, FEh = –2 = divided by 2)
Multiplication ratio 2 (1 byte):	Multiplication/division ratio of the input frequency to obtain the peripheral clock (PCLK) This value is represented in the same format as multiplication ratio 1
SUM (1 byte):	Checksum
Error:	Error code 11h: Checksum error 24h: Bit rate selection error 25h: Input frequency error 26h: Multiplication ratio error 27h: Operating frequency error

- Bit rate selection error

A bit rate selection error occurs when the bit rate selected through a new bit rate selection command cannot be set for the SCI of the RX63N/RX631 within an error of 4%. The bit rate error can be obtained by the following equation from the bit rate (B) selected through a new bit rate selection command, the input frequency (f_{EX}), multiplication ratio 2 ($M_{P\phi}$), the bit rate register (BRR) setting (N) in the SCI, and the CKS[1:0] bit value (n) in the serial mode register (SMR).

$$\text{Error (\%)} = \frac{f_{EX} \times M_{P\phi} \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1$$

- Input frequency error

An input frequency error occurs when the input frequency specified through a new bit rate selection command is outside the range from the minimum to maximum input frequencies for the clock mode selected through a clock mode selection command.

- Multiplication ratio error

A multiplication ratio error occurs when the multiplication ratio specified through a new bit rate selection command does not match the clock mode selected through a clock mode selection command. To check the selectable multiplication ratios, issue a multiplication ratio inquiry command.

- Operating frequency error

An operating frequency error occurs when the RX63N/RX631 cannot operate at the operating frequencies selected through a new bit rate selection command. The RX63N/RX631 calculates the operating frequencies from the input frequency and multiplication ratios specified through a new bit rate selection command and checks if each calculated frequency is within the range from the minimum to maximum frequencies for the respective clock. To check the minimum and maximum operating frequencies for each clock, issue an operating frequency inquiry command.

(12) Programming/Erasure State Transition

In response to a programming/erasure state transition command sent from the host, the RX63N/RX631 determines whether ID code protection is enabled or disabled using the control code and ID code written in the ROM. When ID code protection is enabled, the RX63N/RX631 returns a response (16h) and waits for the ID code. When ID code protection is disabled, the RX63N/RX631 erases the entire area of each of the user area, user boot area and data area. After completing entire erasure, the RX63N/RX631 returns a response (26h) and waits for a programming/erasure host command. If the RX63N/RX631 has failed to complete erasure due to an error, it returns an error response (sends C0h and 51h in that order).

Do not issue a programming/erasure state transition command before the device selection, clock mode selection, and new bit rate selection commands.

Command	40h	
Response	ACK	
Error response	C0h	51h

ACK (1 byte): ACK code
26h: ID code protection is disabled
16h: ID code protection is enabled

(13) Embedded Program Status Inquiry

In response to an embedded program status inquiry command sent from the host, the RX63N/RX631 returns its current status. The embedded program status inquiry command can be issued in both the inquiry/selection host command wait state and programming/erasure host command wait state.

Command	4Fh			
Response	5Fh	Size	Status	Error

Size (1 byte): Total number of bytes in the status and error fields (fixed at 2)
Status (1 byte): Current status of RX63N/RX631 (see Table 45.13)
Error (1 byte): Error status of RX63N/RX631 (see Table 45.14)

Table 45.13 Status Code

Code	Description
11h	Waiting for device selection
12h	Waiting for clock mode selection
13h	Waiting for bit rate selection
1Fh	Waiting for transition to programming/erasure host command wait state (bit rate has been selected)
31h	Erasing the user area/user boot area
3Fh	Waiting for a programming/erasure host command
4Fh	Waiting for reception of programming data
5Fh	Waiting for erasure block specification

Table 45.14 Error Code

Code	Description
00h	No error
11h	Checksum error
21h	Incorrect device code error
22h	Incorrect clock mode error
24h	Bit rate selection error
25h	Input frequency error
26h	Multiplication ratio error
27h	Operating frequency error
29h	Block number error
2Ah	Address error
2Bh	Data size error
51h	Erasure error
52h	Incomplete erasure error
53h	Programming error
54h	Selection error
80h	Command error
FFh	Bit rate adjustment confirmation error

45.10.7 ID Code Wait State

Table 45.15 shows the host command available in the ID code wait state.

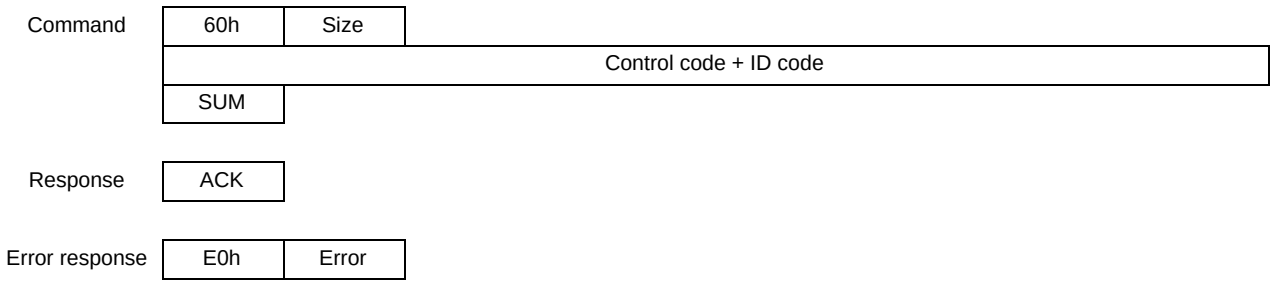
Table 45.15 ID Code Check Host Command

Host Command Name	Function
ID code check	Performs the ID code check

If the host has sent an undefined command, the RX63N/RX631 returns a response indicating a command error. For the contents of a command error, see section 45.10.6, Inquiry/Selection Host Command Wait State.

(1) ID Code Check

In response to an ID code check command sent from the host, the RX63N/RX631 compares the code sent from the host with the control code and ID code in the ROM and returns the result.



Size (1 byte): Number of bytes in the ID code field (fixed at 16)

ID code (16 bytes): Control code (1 byte) + ID code (15 bytes)

SUM (1 byte): Checksum

ACK (1 byte): ACK code
26h: Returns the response for a programming/erasure state transition command

Error (1 byte): Error code
11h: Checksum error
61h: ID code mismatch
63h: ID code mismatch (erasure error)
An error has occurred during erasure triggered by an ID code mismatch.

45.10.8 Programming/Erasure Host Command Wait State

Table 45.16 lists the host commands available in the programming/erasure host command wait state.

Table 45.16 Programming/Erasure Host Commands

Host Command Name	Function
User boot area programming selection	Selects the program for user boot area programming
User area programming selection	Selects the program for user area programming
256-byte programming	Programs 256 bytes of data
Erasure selection	Selects the erasure program
Block erase	Erases block data
Memory read	Reads data from memory
User boot area checksum	Performs checksum verification for the user boot area
User area checksum	Performs checksum verification for the user area
User boot area blank check	Checks whether the user boot area is blank
User area blank check	Checks whether the user area is blank
Read lock bit status	Reads from the lock bit
Lock bit program	Writes to the lock bit
Lock bit enable	Enables the lock bit protection
Lock bit disable	Disables the lock bit protection
Embedded program status inquiry	Inquires regarding the state of the RX63N/RX631

If the host has sent an undefined command, the RX63N/RX631 returns a response indicating a command error. For the contents of a command error, see section 45.10.6, Inquiry/Selection Host Command Wait State.
To program the ROM, issue a programming selection command (user area programming selection/user boot area programming selection) and then a 256-byte programming command from the host. Upon reception of a programming selection command, the RX63N/RX631 enters the programming data wait state (see section 45.10.2, State Transitions in Boot Mode). In response to a 256-byte programming command sent from the host in this state, the RX63N/RX631 starts programming the ROM. When the host sends a 256-byte programming command specifying FFFF FFFFh as the programming start address, the RX63N/RX631 detects it as the end of programming and enters the programming/erasure host command wait state.
To erase the ROM, issue an erasure selection command and then a block erase command from the host. Upon reception of an erasure selection command, the RX63N/RX631 enters the erasure block selection wait state (see section 45.10.2, State Transitions in Boot Mode). In response to a block erase command sent from the host in this state, the RX63N/RX631 erases the specified block in the ROM. When the host sends a block erase command specifying FFh as the block number, the RX63N/RX631 detects it as the end of erasure and enters the programming/erasure host command wait state.

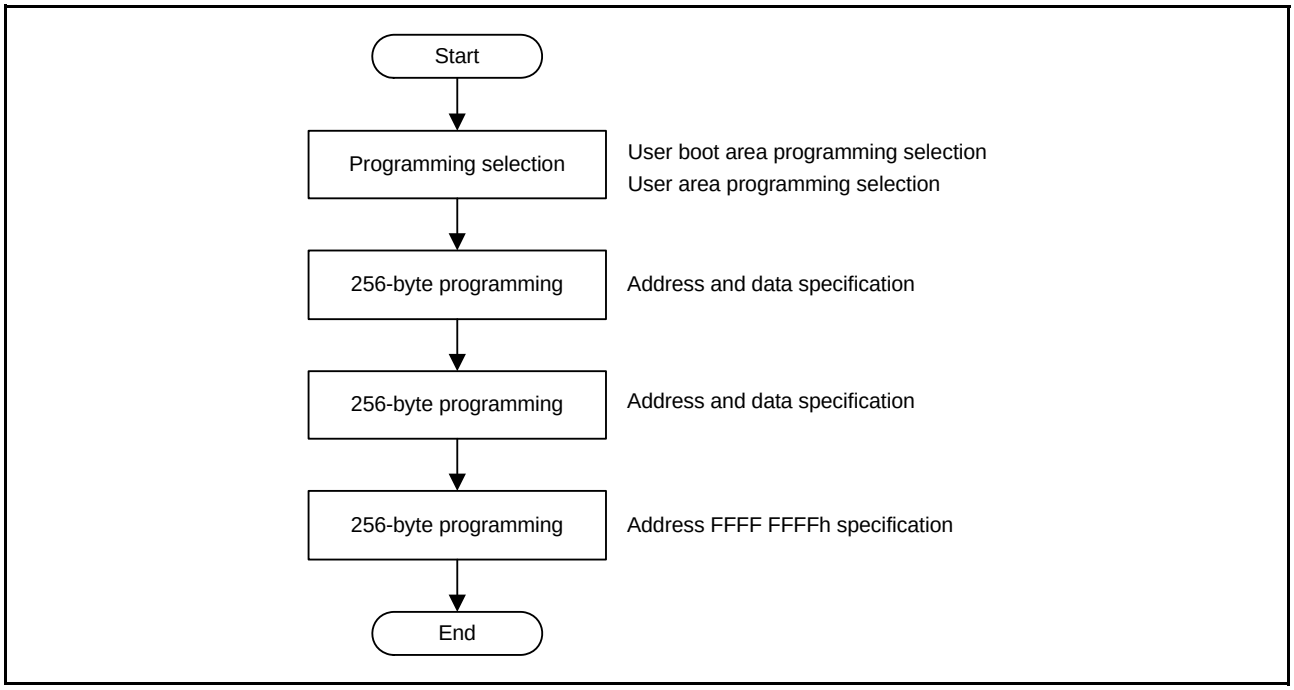


Figure 45.30 Procedure for ROM Programming in Boot Mode

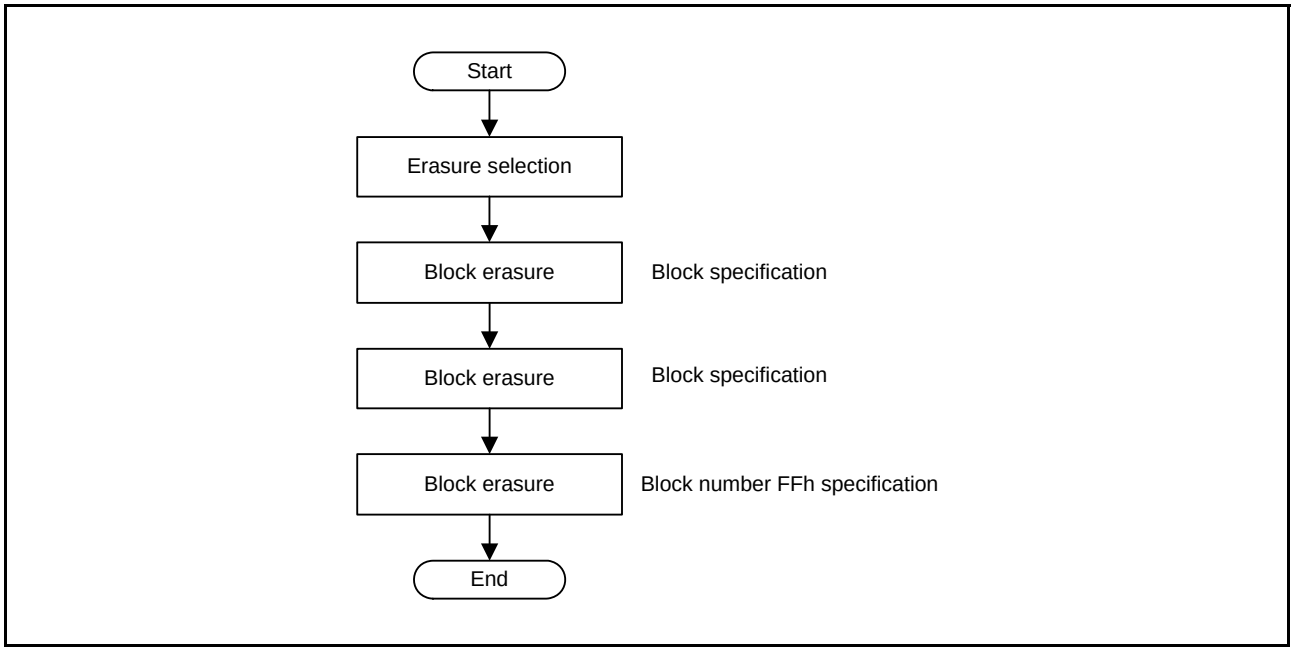


Figure 45.31 Procedure for ROM Erasure in Boot Mode

Each host command is described in detail below. The “command” in the description indicates a command sent from the host to the RX63N/RX631 and the “response” indicates a response sent from the RX63N/RX631 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX63N/RX631 becomes 00h.

(1) User Boot Area Programming Selection

In response to a user boot area programming selection command sent from the host, the RX63N/RX631 selects the program for user boot area programming and waits for programming data.

Command	42h
Response	06h

(2) User Area Programming Selection

In response to a user area programming selection command sent from the host, the RX63N/RX631 selects the program for user area programming and waits for programming data.

Command	43h
Response	06h

(3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, the RX63N/RX631 programs the ROM. After completing ROM programming successfully, the RX63N/RX631 returns a response (06h). If an error has occurred during ROM programming, the RX63N/RX631 returns an error response (D0h).

Command	50h	Programming address		
	Data	Data	. . .	Data
	SUM			
Response	06h			
Error response	D0h	Error		

Programming address (4 bytes):	Target address of programming To program the ROM, a 256-byte boundary address should be specified. To terminate programming, FFFF FFFFh should be specified.
Data (256 bytes):	Programming data FFh should be specified for the bytes that do not need to be programmed. When terminating programming, no data needs to be sent (only the programming address and SUM should be sent in that order).
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error (the specified address is not in the target area) 53h: Programming cannot be done due to a programming error

(4) Erasure Selection

In response to an erasure selection command sent from the host, the RX63N/RX631 selects the erasure program and waits for erasure block specification.

Command	48h
Response	06h

(5) Block Erasure

In response to a block erase command sent from the host, the RX63N/RX631 erases the ROM. After completing ROM erasure successfully, the RX63N/RX631 returns a response (06h). If an error has occurred during ROM erasure, the RX63N/RX631 returns an error response (D8h).

Command	58h	Size	Block	SUM
Response	06h			
Error response	D8h	Error		

Size (1 byte): Number of bytes in the block specification field (fixed at 1)

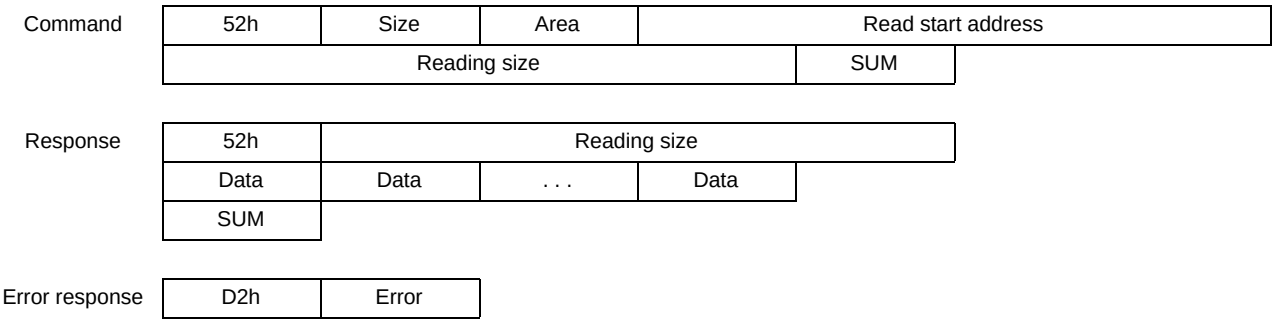
Block (1 byte): Block number whose data is to be erased
To terminate erasure, FFh should be specified.

SUM (1 byte): Checksum

Error (1 byte): Error code
11h: Checksum error
29h: Block number error (an incorrect block number is specified)
51h: Erasure cannot be done due to an erasure error

(6) Memory Read

In response to a memory read command sent from the host, the RX63N/RX631 reads data from the ROM. After completing ROM reading successfully, the RX63N/RX631 returns the data stored in the address specified by the memory read command. If the RX63N/RX631 has failed to read the ROM, the RX63N/RX631 returns an error response (D2h).



Size (1 byte):	Total number of bytes in the area, read start address, and reading size fields
Area (1 byte):	Target area to be read 00h: User boot area 01h: User area
Read start address (4 bytes):	Start address of the area to be read
Reading size (4 bytes):	Size of data to be read (bytes)
SUM (1 byte):	Checksum
Data (1 byte):	Data read from the ROM
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error • The value specified for area selection is neither 00h nor 01h. • The specified read start address is outside the selected area. 2Bh: Data size error • 00h is specified for the reading size. • The reading size is larger than the area. • The end address calculated from the read start address and the reading size is outside the selected area.

(7) User Boot Area Checksum

In response to a user boot area checksum command sent from the host, the RX63N/RX631 sums the user boot area data in byte units and returns the result (checksum).

Command	4Ah			
Response	5Ah	Size	Area checksum	SUM

Size (1 byte):	Number of bytes in the area checksum field (fixed at 4)
Area checksum (4 bytes):	Checksum of the user boot area data
SUM (1 byte):	Checksum (for the response data)

(8) User Area Checksum

In response to a user area checksum command sent from the host, the RX63N/RX631 sums the user area data in byte units and returns the result (checksum).

Command	4Bh			
Response	5Bh	Size	Area checksum	SUM

Size (1 byte):	Number of bytes in the area checksum field (fixed at 4)
Area checksum (4 bytes):	Checksum of the user area data The user area also stores the key code for debugging function authentication. Note that the checksum includes this key code value.
SUM (1 byte):	Checksum (for the response data)

(9) User Boot Area Blank Check

In response to a user boot area blank check command sent from the host, the RX63N/RX631 checks whether the user boot area is completely erased. When the user boot area is completely erased, the RX63N/RX631 returns a response (06h). If the user boot area has an unerased area, the RX63N/RX631 returns an error response (sends CDh and 52h in that order).

Command	4Ch	
Response	06h	
Error response	CCh	52h

(10) User Area Blank Check

In response to a user area blank check command sent from the host, the RX63N/RX631 checks whether the user area is completely erased. When the user area is completely erased, the RX63N/RX631 returns a response (06h). If the user area has an unerased area, the RX63N/RX631 returns an error response (sends CDh and 52h in that order).

Command	4Dh	
Response	06h	
Error response	CDh	52h

(11) Read Lock Bit Status

In response to a read lock bit status command sent from the host, the RX63N/RX631 reads data from the lock bit. After completing the lock bit reading successfully, the RX63N/RX631 returns the data stored in the address specified by the read lock bit status command. If the RX63N/RX631 has failed to read the lock bit, the RX63N/RX631 returns an error response (F1h).

Command	71h	Size	Area	Third highest order address	Second highest order address	Highest order address	SUM
Response	Status						
Error response	F1h	Error					

Size (1 byte):	Total number of bytes in the area, third highest order address, second highest order address, and highest order address fields (fixed at 4 in the RX63N/RX631)
Area (1 byte):	Target area to be read 01h: User area
Third highest order address (1 byte):	Third highest order address at the specified block's end address (8 to 15 bits)
Second highest order address (1 byte):	Second highest order address at the specified block's end address (16 to 23 bits)
Highest order address (1 byte):	Highest order address at the specified block's end address (24 to 31 bits)
SUM (1 byte):	Checksum
Status (1 byte):	Bit 6 locked at 0 Bit 6 unlocked at 1
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error (the specified address is not in the target area)

(12) Lock Bit Program

In response to a lock bit program command sent from the host, the RX63N/RX631 writes to a lock bit and locks the specified block. After completing the lock bit blocking successfully, the RX63N/RX631 returns a response (06h). If the RX63N/RX631 has failed to lock, the RX63N/RX631 returns an error response (F7h).

Command	77h	Size	Area	Third highest order address	Second highest order address	Highest order address	SUM
Response	06h						
Error response	F7h	Error					

Size (1 byte):	Total number of bytes in the area, third highest order address, second highest order address, and highest order address fields (fixed at 4 in the RX63N/RX631)
Area (1 byte):	Target area to be locked 01h: User area
Third highest order address (1 byte):	Third highest order address at the specified block's end address (8 to 15 bits)
Second highest order address (1 byte):	Second highest order address at the specified block's end address (16 to 23 bits)
Highest order address (1 byte):	Highest order address at the specified block's end address (24 to 31 bits)
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error (the specified address is not in the target area) 53h: Lock cannot be done due to a programming error

(13) Lock Bit Enable

In response to a lock bit enable command sent from the host, the RX63N/RX631 enables a lock bit.

Command	7Ah
Response	06h

(14) Lock Bit Disable

In response to a lock bit disable command sent from the host, the RX63N/RX631 disables a lock bit.

Command	75h
Response	06h

(15) Embedded Program Status Inquiry

For details, refer to section 45.10.6, Inquiry/Selection Host Command Wait State.

45.11 USB Boot Mode

In USB boot mode, the user area is programmed or erased by control commands and data for programming transmitted from an externally connected host via the USB.
Using USB boot mode requires preparation on the host side of tools for transmitting the control commands and data for programming, and of the data. Figure 45.32 shows the configuration of a system for use in USB boot mode. Interrupt requests generated in USB boot mode are ignored. Ensure that interrupt requests are not generated on the system side.

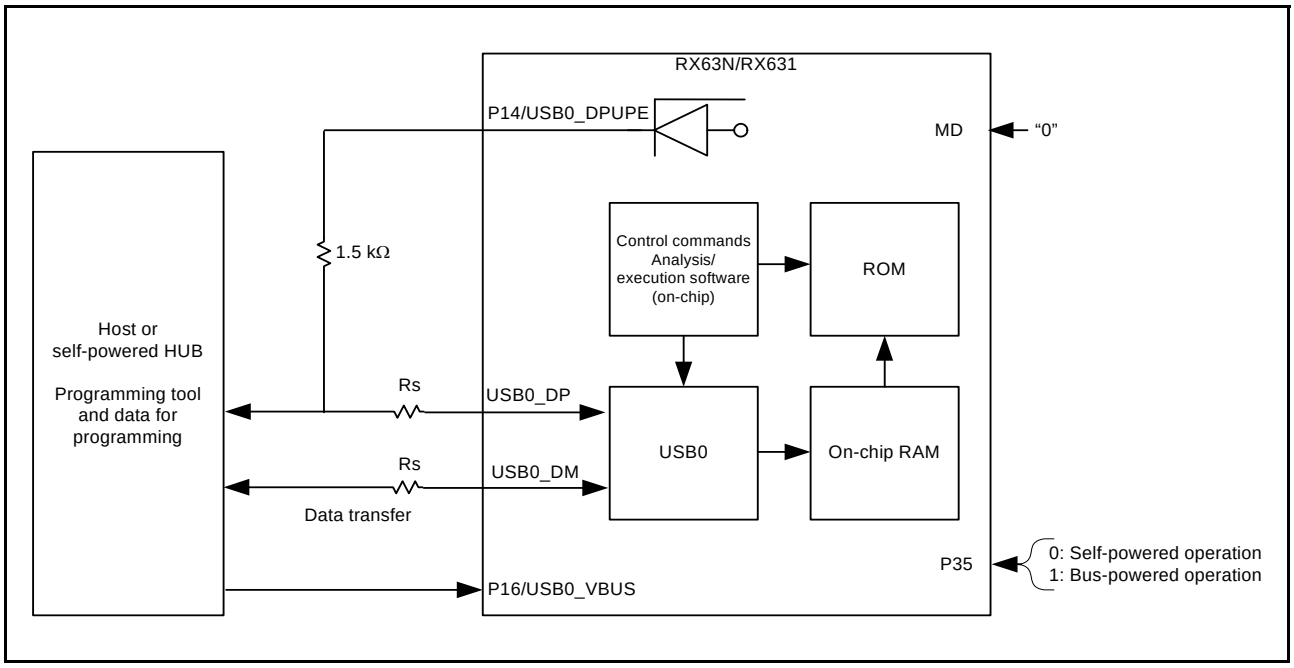


Figure 45.32 System Configuration in USB Boot Mode

45.11.1 Features

- Selection of bus-powered or self-powered mode
- Only the USB0_DPUPE pin is used for control of D+ pull-up connection
- See Table 45.17 for the enumeration information

Table 45.17 Enumeration Information

USB specification	Ver. 2.0 (Full-speed)	
Transfer modes	Control (in/out), Bulk (in/out)	
Maximum current	Self-powered mode (pin P35 = 0)	100 mA
	Bus-powered mode (pin P35 = 1)	500 mA
Endpoint configuration	EP0 Control (in out) 8 Bytes Configuration1 └─InterfaceNumber0 └─AlternateSetting0 ├─EP1 Bulk (out) 64Bytes └─EP2 Bulk (in) 64Bytes	

45.11.2 State Transitions

State transitions after activation in USB boot mode are shown in Figure 45.33.

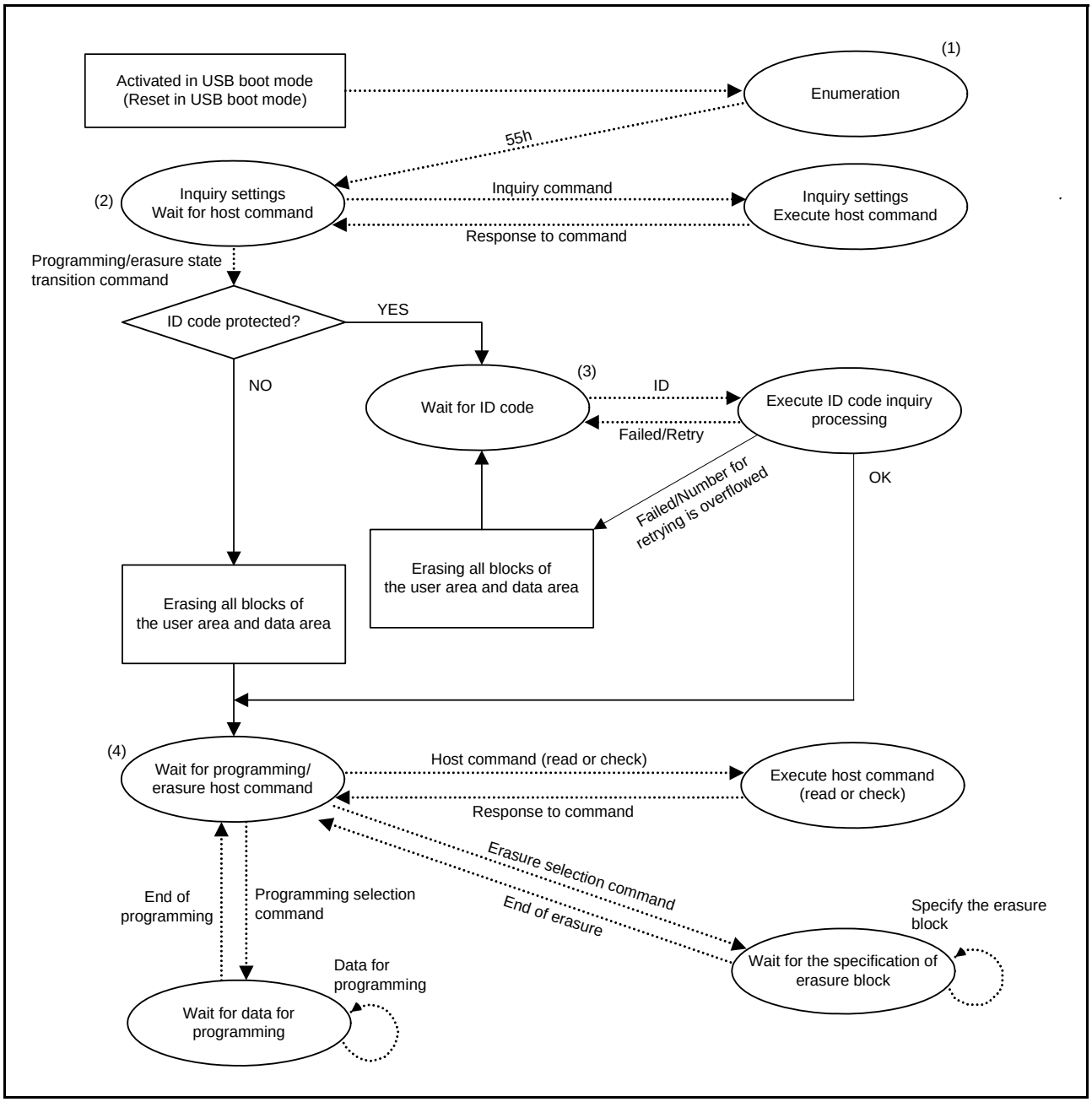


Figure 45.33 State Transitions in USB Boot Mode

(1) On the transition to USB boot mode, the boot program embedded within this chip is initiated. When the boot program is initiated, the process of enumeration is conducted with the host. Once enumeration is completed, the host sends a single byte (55h) to this chip. If this byte is not received correctly, booting up in USB boot mode is restarted.
(2) Inquiry data on the size, structure, first address, state of support, etc. of the user area are sent to the host.
(3) On completion of the inquiries, the whole user area is automatically erased.
(4) After automatic erasure of the user area, the state shifts to waiting for programming and erasure commands. When a programming command is received, the state shifts to waiting for data to be programmed. When an erasure-selection command is received, the state shifts to waiting for specification of blocks to be erased.
Other than the programming and erasure commands, commands for the following processes on the user area are also possible: sum checking, blank checking (checking erasure), reading memory, and acquiring current status information.

45.11.3 Notes on Program Execution in USB Boot Mode

(1) A 48-MHz clock signal must be provided to the USB module. Set the clock pulse generator and external clock frequency to ensure that the dedicated clock for the USB (UCLK) is at 48 MHz. For details, refer to section 9, Clock Generation Circuit.
(2) Use the USB0_DPUPE pin for controlling D+ pull-up connection.
(3) To ensure that the power supply is stable during programming and erasure of flash memory, do not connect a cable via a bus-powered HUB.
(4) Disconnection of the USB cable during programming and erasure of flash memory can permanently damage the LSI, so take particular care.
(5) Even if the USB bus enters the suspended mode in bus-powered mode, this LSI will not enter software standby mode (low-power consumption mode).

45.12 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 ROM 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 ROM 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. However, if the control code is 52h and 50h, 72h, 6Fh, 74h, 65h, 63h, 74h are set to ID codes 1 to 7, respectively there is no determination of matching and the ID code is always considered to be non-matching, and connection of the on-chip debugger is prohibited. Furthermore, if all bytes of the control code and ID code have the value FFh, there is no determination of matching, the ID code is always considered to match, and connection of the on-chip debugger is allowed. See Figure 45.27 for the configuration of ID codes in flash memory.

Table 45.18 Specifications for ID Code Protection on Connection of the On-Chip Debugger

Control Code	ID Code	State of Protection	Operations at On-Chip Debugger Connection
FFh	FFh, …, FFh (all bytes FFh)	Protection disabled	The ID code always matches, and connection to the on-chip debugger is permitted.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h	Protection enabled	The ID code is always non-matching, and connection to the on-chip debugger is prohibited.
Other than above	Other than above	Protection enabled	Matching ID code: Authentication of the on-chip debugger is ended and connection with the on-chip debugger is permitted. Non-matching ID code: The ID code protection waiting state is entered again.

45.13 ROM Code Protection

ROM code protection is a facility for prohibiting a PROM programmer from reading from or programming flash memory. The ROM code in flash memory is a 32-bit code. Figure 45.34 shows the configuration of the ROM code. Set the ROM code in 32-bit units.
For release from ROM code protection, erase the EB00 block of the user area that contains the ROM code in boot mode or by user programming.

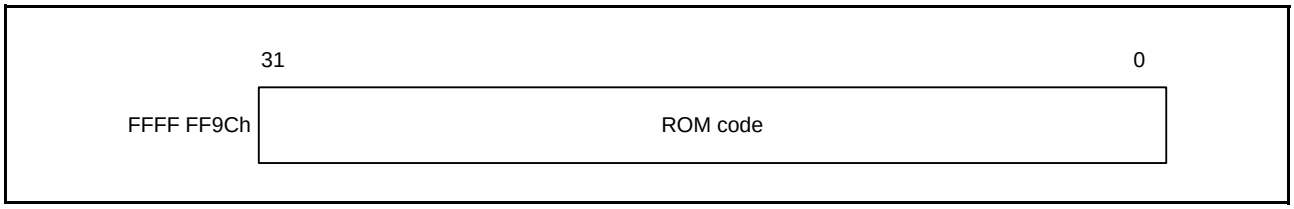


Figure 45.34 Configuration of ROM Code

Table 45.19 Specifications for ROM Code Protection

ROM Code	State of Protection	Operations at the Time of Connection with the PROM Programmer
0000 0000h	Protection enabled (ROM code protection 1)	Access (both reading and writing) 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 writing) to the user area and the user boot area are permitted.

45.14 Usage Notes

(1) Areas where Programming or Erasure is Suspended

Data in areas where programming or erasure is suspended are undefined. To avoid malfunctions due to the reading of undefined data, prevent the reading of data and execution of code from areas where programming or erasure is currently suspended.

(2) Suspending Programming or Erasure

If you use the programming/erasure suspension command to suspend the processing of programming or erasure, be sure to use the resume command so that the processing is completed.

(3) Prohibition of Reprogramming

Two or more programming operations cannot be performed for the same address range. If an address range that has already been programmed is to be programmed again, be sure to erase the area in advance of the programming.

(4) Reset during Programming or Erasure

Do not allow the generation of any reset (by the signal on the RES# pin, a voltage-monitoring 0 reset, a voltage-monitoring 1 reset, a voltage-monitoring 2 reset, an IWDT reset, a watchdog-timer reset, or a software reset) during programming and erasure of the flash memory, as this may damage the flash memory. When a reset by the signal on the RES# pin must be input, only release the chip from the reset state after at least tRESWF (see section 49., Electrical Characteristics) has elapsed.
When the FCU is reset by the FRESETR.FRESET bit during programming/erasure, make sure that the reset state is maintained for the period of tFCUR (see section 49., Electrical Characteristics). Do not attempt to read the ROM during a reset.

(5) Prohibition of Non-Maskable Interrupts during Programming or Erasure

If a non-maskable interrupt (NMI pin interrupt, oscillation stop detection interrupt, WDT underflow, refresh error, IWDT underflow/refresh error, voltage-monitoring 1 interrupt or voltage-monitoring 2 interrupt) occurs during programming or erasure, as this will lead to fetching of the vector from the ROM, and the data read out will be undefined. Therefore, avoid a maskable interrupt being generated during programming or erasing in the ROM.

(6) Interrupt Vector Assignment During Programming or Erasure

The generation of interrupts during programming or erasure may lead to the fetching of vectors from the ROM. To prevent access to the ROM area due to the generation of interrupts, set the interrupt table register (INTB) of the CPU so that the destination for the fetching of interrupt vector is an area outside the ROM.

(7) Programming and Erasure in Low-Speed Operating Modes 1 and 2

Do not program or erase the flash memory after low-speed operating mode 1 or 2 has been selected in the operating voltage control register (OPCCR).

(8) Programming/Erasure Abnormal End

If programming/erasure is not successfully completed due to a reset of programming/erasure, a reset through the FRESETR.FRESET or a command-locked state caused by error detection, the lock bit may be 0 (protected state). In this case, erase the lock bit by issuing the block erase command with the FPROTR.FPROTCN bit set to 1.

(9) Actions Prohibited during Programming and Erasure

Programming and erasure lead to the application of high voltages within flash memory. The following operations are prohibited to prevent damage to flash memory.

- The power supply of the RX63N/RX631 is at or below the operating voltage.
- The value of the FWEPROR.FLWE[1:0] bits is changed.
- The setting of the SYSCR0.ROME bit changes the operating mode.
- The value of the OPCCR.OPCM[2:0] bits is changed.
- The clock-source selection bits in register SCKCR3 are changed.
- Switching of the clock source for use on recovery from sleep mode is enabled by the setting of the RSTCKCR.RSTCKEN bit.
- The frequency dividing ratio for the FlashIF clock (FCLK) is changed.
- Transition to all-module clock-stop mode, software standby or deep software standby

(10) Notes on Flash Programming in Boot Mode or USB Boot Mode

The main clock must be being input for programming of the flash memory in boot mode or USB boot mode. In the case of boot mode, a crystal oscillator within the range indicated in section 45.4, Clock Timing, must be connected between the XTAL and EXTAL pins. Connect a 12-MHz oscillator if operation is in USB boot mode.

(11) Handling of the EXTAL Pin in Boot Mode

When operation is in boot mode, provide a clock signal through connection of an external input or crystal oscillator to the EXTAL pin.

(12) Handling of the EXTAL Pin in USB Boot Mode

When operation is in USB boot mode, provide a 12-MHz clock signal through connection of a crystal oscillator to the EXTAL pin.

46. E2 DataFlash Memory (Flash Memory for Data Storage)

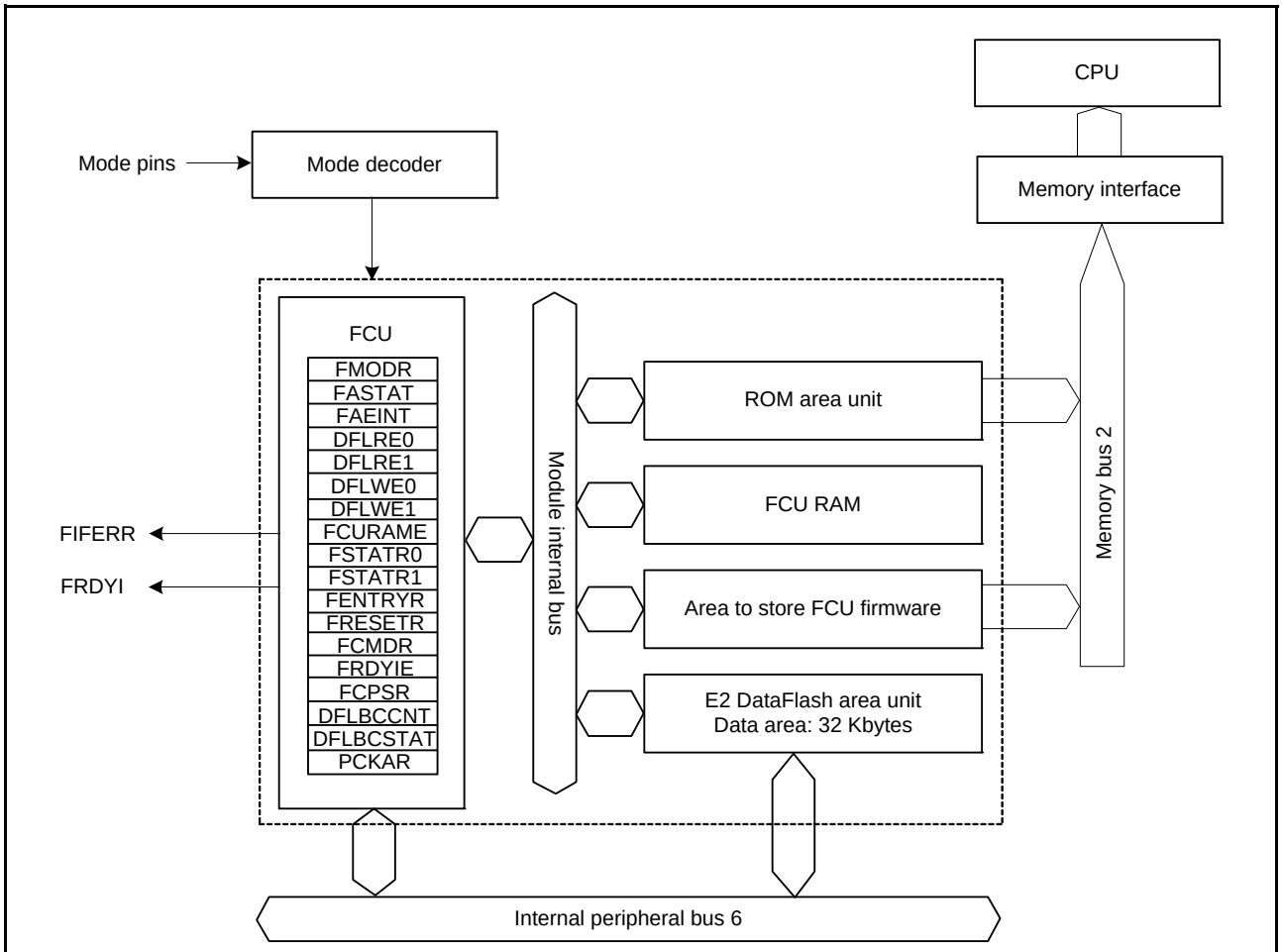
The RX63N/RX631 Group has a 32-Kbyte flash memory for storing data (E2 DataFlash).
This section covers the E2 DataFlash memory. For the ROM, see section 45, ROM (Flash Memory for Code Storage).

46.1 Overview

Table 46.1 lists the specifications of the E2 DataFlash memory, and Figure 46.1 is a block diagram of the ROM, E2 DataFlash memory, and related modules.

Table 46.1 Specifications of E2 DataFlash Memory

Item		Specifications
Memory capacity		Data area: 32 Kbytes
Reading via the peripheral bus		A read operation takes six cycles of FCLK in words or bytes
Programming/erasing method		• The chip incorporates a dedicated sequencer (FCU) for programming of the ROM and E2 DataFlash. • Programming and erasing the ROM and E2 DataFlash are handled by issuing commands to the FCU.
BGO (background operation)		• Execution of program code from the ROM is possible while the E2 DataFlash memory is being programmed or erased. • The CPU is able to execute program code from areas other than the ROM or E2 DataFlash while the ROM is being programmed or erased.
Suspension and resumption		• The CPU is able to execute program code from the E2 DataFlash during suspension of programming or erasure. • Programming and erasure of the E2 DataFlash can be restarted (resumed) after suspension.
Units of programming and erasure		• Unit of programming for the data area: 2 bytes • Unit of erasure for the data area: 32 bytes (1024 blocks)
Blank checking function		• The blank checking command can be executed to check the erasure state of E2 DataFlash. • The size of the area to be blank-checked is 2 bytes or 2 Kbytes.
On-board programming (four types)	Boot mode	• The data area is programmable via the SCI. • The bit rate for SCI transfer between the host and RX63N/RX631 is automatically adjusted.
	USB boot mode	The data area is programmable via the USB.
	User boot mode	The data area can be programmed after the system is started using the user boot area.
	User program mode	Programming of the data area under program control
Protection	Software-controlled protection	• The FENTRYR.FENTRYD bit, FWEPROR.FLWE[1:0] bits, DFLREy, and DFLWEy register (y = 0, 1), can be used to prevent unintentional programming or reading. • Protection with the DFLREn and DFLWEn registers is performed on a 2-Kbyte basis.
	Error protection	Prevention of further programming or erasure after the detection of abnormal operations during programming or erasure
Times for programming and erasure, durability (number of times reprogramming is possible)		See section 45, Electrical Characteristics.



FMODR:	Flash mode register
FASTAT:	Flash access status register
FAEINT:	Flash access error interrupt enable register
DFLRE0:	E2 DataFlash read enable register 0
DFLRE1:	E2 DataFlash read enable register 1
DFLWE0:	E2 DataFlash programming/erasure enable register 0
DFLWE1:	E2 DataFlash programming/erasure enable register 1
FCURAME:	FCU RAM enable register
FSTATR0:	Flash status register 0
FSTATR1:	Flash status register 1
FENTRYR:	Flash P/E mode entry register
FRESETR:	Flash reset register
FCMDR:	FCU command register
FRDYIE:	Flash ready interrupt enable register
FCPSR:	FCU process switch register
DFLBCCNT:	E2 DataFlash blank check control register
DFLBCSTAT:	E2 DataFlash blank check status register
PCKAR:	Peripheral clock notification register
FIFERR:	Flash interface error interrupt
FRDYI:	Flash ready interrupt

Figure 46.1 Block Diagram of E2 DataFlash Memory

Input and output pins associated with the E2 DataFlash are listed in Table 46.2.

Table 46.2 Input and Output Pins Associated with the E2 DataFlash

Pin Name	I/O	Description
PF2/RXD1 (177/176-pin packages) P30/RXD1 (145/144/100-pin packages)	Input	Used in boot mode to receive data via SCI (for host communications)
PF0/TXD1 (177/176-pin packages) P26/TXD1 (145/144/100-pin packages)	Output	Used in boot mode to transmit data from SCI (for host communications)
MD	Input	Operating mode settings
PC7	Input	Selection of boot mode (SCI boot), USB boot mode or user boot mode
USB0_DP, USB0_DM	I/O	Input/output pins for USB data (for use in USB boot mode)
P14/USB0_DPUPE	I/O	Pull-up pin for USB (for use in USB boot mode)
P16/USB0_VBUS	Input	Detection of connection and disconnection of USB cables (for use in USB boot mode)
P35	Input	Selection of USB bus-power mode or self-power mode (for use in USB boot mode)

46.2 Register Descriptions

Some registers also have bits related to the ROM, but this section deals only with the bits that are relevant to the E2 DataFlash. For registers containing bits with common functions for the ROM and E2 DataFlash (FRDYIE, FCURAME, FSTATR0, FSTATR1, FRESETR, FCMDR, FCPSR, PCKAR, and FWEPROR) and details on the functions of bits dedicated to the ROM, see section 45.2, Register Descriptions.
P/E indicates programming/erasure.

46.2.1 Flash Mode Register (FMODR)

Address(es): 007F C402h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FRDMD	—	—	—	—
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	FRDMD	FCU Read Mode Select	0: Memory Area Reading Method This is the setting when a transition of E2 DataFlash lock bit read mode is made. Since there are no lock bits for the E2 DataFlash, undefined data are read from the E2 DataFlash area after a transition of the lock bit reading mode is made. 1: Register Reading Method This is the setting when the blank checking command is to be used.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set the FRDMD bit to 1 if the blank checking command is to be used.
In modes in which the on-chip ROM is disabled, the value read from FMODR is 00h and writing is disabled.

FRDMD Bit (FCU Read Mode Select)

This bit is used to select processing for a transition of E2 DataFlash lock bit read mode or for blank checking.
The FRDMD bit is used to select the method of reading when lock bit values for the ROM are read (see section 45, ROM (Flash Memory for Code Storage)).

46.2.2 Flash Access Status Register (FASTAT)

Address(es): 007F C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAE	—	—	CMDLK	DFLAE	—	DFLRPE	DFLWPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPE	E2 DataFlash Programming/Erasure Protection Violation	0: An E2 DataFlash programming/erasure command is not issued which conflicts with the DFLWEy settings 1: An E2 DataFlash programming/erasure command is issued which conflicts with the DFLWEy settings (y = 0, 1)	R/(W) *1
b1	DFLRPE	E2 DataFlash Read Protection Violation	0: There is no such E2 DataFlash reading that conflicts with the DFLREy settings 1: There is such an E2 DataFlash reading that conflicts with the DFLREy settings (y = 0, 1)	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DFLAE	E2 DataFlash Access Violation	0: No E2 DataFlash access violation 1: E2 DataFlash access violation	R/(W) *1
b4	CMDLK	FCU Command Lock	0: FCU is not in the command-locked state 1: FCU is in the command-locked state	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAE	ROM Access Violation	See section 45, ROM (Flash Memory for Code Storage).	R/(W) *1

Note 1. Only 0 can be written after reading 1 to clear the flag.

When on-chip ROM is disabled, the data read from FASTAT is 00h and writing is disabled. When one of the bits in FASTAT is set to 1, the FCU is placed in the command-locked state (see section 46.7.2, Error Protection). To clear the command-locked state, a status register clearing command must be issued to the FCU after setting FASTAT to 10h.

DFLWPE Bit (E2 DataFlash Programming/Erasure Protection Violation)

This bit is used to indicate whether or not the programming/erasure protection set by DFLWEy (y = 0, 1) is violated. When the DFLWPE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, placing the FCU in the command-locked state. For FSTATR0, see section 45.2.6, Flash Status Register 0 (FSTATR0).

[Setting condition]

- A programming/erasure command is issued for an E2 DataFlash area for which programming or erasure is disabled by DFLWEy.

[Clearing condition]

- When 0 is written after reading 1

DFLRPE Bit (E2 DataFlash Read Protection Violation)

This bit is used to indicate whether or not the reading protection set by DFLREy (y = 0, 1) is violated.
When the DFLRPE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, placing the FCU in the command-locked state.
For FSTATR0, see section 45.2.6, Flash Status Register 0 (FSTATR0).

[Setting condition]

- A read command is issued for an E2 DataFlash area for which reading is disabled by DFLREy.

[Clearing condition]

- When 0 is written after reading 1

DFLAE Bit (E2 DataFlash Access Violation)

This bit indicates whether an E2 DataFlash access violation occurred.
When the DFLAE bit is set to 1, the ILGLERR bit in FSTATR0 is set to 1, placing the FCU in the command-locked state.
For FSTATR0, see section 45.2.6, Flash Status Register 0 (FSTATR0).

[Setting conditions]

- A read command is issued for an E2 DataFlash area in E2 DataFlash P/E normal mode and when the FENTRYD bit in FENTRYR is set to 1.
- A write command is issued for an E2 DataFlash area when the FENTRYD bit is set to 0.
- A command is issued for an E2 DataFlash area when any of bits FENTRY3, FENTRY2, FENTRY1, and FENTRY0 in FENTRYR is set to 1.

[Clearing condition]

- When 0 is written after reading 1

CMDLK Bit (FCU Command-Locked)

This bit indicates that the FCU is in the command-locked state (see section 46.7.2, Error Protection).

[Setting condition]

- After the FCU detects an error and enters the command-locked state

[Clearing condition]

- After the FCU has processed a status register clearing command

46.2.3 Flash Access Error Interrupt Enable Register (FAEINT)

Address(es): 007F C411h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
Value after reset:	1	0	0	1	1	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPEIE	E2 DataFlash Programming/Erasure Protection Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the DFLWPE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLWPE bit in FASTAT is set to 1	R/W
b1	DFLRPEIE	E2 DataFlash Read Protection Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the DFLRPE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLRPE bit in FASTAT is set to 1	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DFLAEIE	E2 DataFlash Access Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the DFLAE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLAE bit in FASTAT is set to 1	R/W
b4	CMDLKIE	FCU Command Lock Interrupt Enable	0: FIFERR interrupt requests disabled when the CMDLK bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the CMDLK bit in FASTAT is set to 1	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAEIE	ROM Access Violation Interrupt Enable	See section 45, ROM (Flash Memory for Code Storage).	R/W

When on-chip ROM is disabled, the data read from FAEINT is 00h and writing is disabled.

DFLWPEIE Bit (E2 DataFlash Programming/Erasure Protection Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash programming/erasure protection violation occurs and the DFLWPE bit in FASTAT is set to 1.

DFLRPEIE Bit (E2 DataFlash Read Protection Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash read protection violation occurs and the DFLRPE bit in FASTAT is set to 1.

DFLAEIE Bit (E2 DataFlash Access Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash access violation occurs and the DFLAE bit in FASTAT is set to 1.

CMDLKIE Bit (FCU Command Lock Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when a FCU command-locked state occurs and the CMDLK bit in FASTAT is set to 1.

46.2.4 E2 DataFlash Read Enable Register 0 (DFLRE0)

Address(es): 007F C440h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00
Value after 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	DBRE00	DB0000-DB0063 (2 Kbytes) Block Read Enable	0: Read disabled 1: Read enabled	R/W
b1	DBRE01	DB0064-DB0127 (2 Kbytes) Block Read Enable		R/W
b2	DBRE02	DB0128-DB0191 (2 Kbytes) Block Read Enable		R/W
b3	DBRE03	DB0192-DB0255 (2 Kbytes) Block Read Enable		R/W
b4	DBRE04	DB0256-DB0319 (2 Kbytes) Block Read Enable		R/W
b5	DBRE05	DB0320-DB0383 (2 Kbytes) Block Read Enable		R/W
b6	DBRE06	DB0384-DB0447 (2 Kbytes) Block Read Enable		R/W
b7	DBRE07	DB0448-DB0511 (2 Kbytes) Block Read Enable		R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable rewriting of the DBREj bit (j = 00 to 07).	R/(W) *1

Note 1. Write data is not retained.

DFLRE0 is a register to enable or disable the DB0000 to DB0511 blocks of the data area (see Figure 46.3) to be read. Reading is enabled or disabled in 2 Kbytes (64 erasure blocks).
Write access to DFLRE0 is valid only when the specific value is written to the upper byte in word access. Data written to the upper byte is not retained.
When on-chip ROM is disabled, the data read from DFLRE0 is 0000h and writing is disabled.

DBREj Bit (DBj Block Read Enable) (j = 00 to 07)

This bit is used to enable or disable the DB0000 to DB0511 blocks of the data area to be read.
The DBREj bit is used to control reading of the DBj blocks.
Writing to the DBREj is enabled only in word access when the KEY[7:0] bits are 2Dh.

46.2.5 E2 DataFlash Read Enable Register 1 (DFLRE1)

Address(es): 007F C442h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
Value after 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	DBRE08	DB0512-DB0575 (2 Kbytes) Block Read Enable	0: Read disabled 1: Read enabled	R/W
b1	DBRE09	DB0576-DB0639 (2 Kbytes) Block Read Enable		R/W
b2	DBRE10	DB0640-DB0703 (2 Kbytes) Block Read Enable		R/W
b3	DBRE11	DB0704-DB0767 (2 Kbytes) Block Read Enable		R/W
b4	DBRE12	DB0768-DB0831 (2 Kbytes) Block Read Enable		R/W
b5	DBRE13	DB0832-DB0895 (2 Kbytes) Block Read Enable		R/W
b6	DBRE14	DB0896-DB0959 (2 Kbytes) Block Read Enable		R/W
b7	DBRE15	DB0960-DB1023 (2 Kbytes) Block Read Enable		R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable rewriting of the DBREj bit (j = 08 to 15).	R/(W) *1

Note 1. Write data is not retained.

DFLRE1 is a register to enable or disable the DB0512 to DB1023 blocks of the data area (see Figure 46.3) to be read. Reading is enabled or disabled in 2 Kbytes (64 erasure blocks).
Write access to DFLRE1 is valid only when the specific value is written to the upper byte in word access. Data written to the upper byte is not retained.
When on-chip ROM is disabled, the data read from DFLRE1 is 0000h and writing is disabled.

DBREj Bit (DBj Block Read Enable) (j = 08 to 15)

This bit is used to enable or disable the DB0512 to DB1023 blocks of the data area to be read.
The DBREj bit is used to control reading of the DBj blocks.
Writing to the DBREj is enabled only in word access when the KEY[7:0] bits are D2h.

46.2.6 E2 DataFlash Programming/Erasure Enable Register 0 (DFLWE0)

Address(es): 007F C450h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								DBWE07	DBW006	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00
Value after 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	DBWE00	DB0000-DB0063 (2 Kbytes) Block Programming/Erasure Enable	0: Programming/erasure disabled 1: Programming/erasure enabled	R/W
b1	DBWE01	DB0064-DB0127 (2 Kbytes) Block Programming/Erasure Enable		R/W
b2	DBWE02	DB0128-DB0191 (2 Kbytes) Block Programming/Erasure Enable		R/W
b3	DBWE03	DB0192-DB0255 (2 Kbytes) Block Programming/Erasure Enable		R/W
b4	DBWE04	DB0256-DB0319 (2 Kbytes) Block Programming/Erasure Enable		R/W
b5	DBWE05	DB0320-DB0383 (2 Kbytes) Block Programming/Erasure Enable		R/W
b6	DBW006	DB0384-DB0447 (2 Kbytes) Block Programming/Erasure Enable		R/W
b7	DBWE07	DB0448-DB0511 (2 Kbytes) Block Programming/Erasure Enable		R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable rewriting of the DBWEj bit (j = 00 to 07).	R/(W) *1

Note 1. Write data is not retained.

DFLWE0 is a register to enable or disable the DB0000 to DB0511 blocks of the data area (see Figure 46.3) to be programmed or erased. Programming or erasing is enabled or disabled in 2 Kbytes (64 erasure blocks).
Write access to DFLWE0 is valid only when the specific value is written to the upper byte in word access. Data written to the upper byte is not retained.
When on-chip ROM is disabled, the data read from DFLWE0 is 0000h and writing is disabled.

DBWEj Bit (DBj Block Programming/Erasure Enable) (j = 00 to 07)

This bit is used to enable or disable the DB0000 to DB0511 blocks of the data area to be programmed or erased.
The DBWEj bit is used to control programming/erasure of the DBj blocks.
Programming of the DBWEj bit is enabled only in word access when the KEY[7:0] bits are 1Eh.

46.2.7 E2 DataFlash Programming/Erasure Enable Register 1 (DFLWE1)

Address(es): 007F C452h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08
Value after 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	DBWE08	DB0512-DB0575 (2 Kbytes) Block Programming/Erasure Enable	0: Programming/erasure disabled 1: Programming/erasure enabled	R/W
b1	DBWE09	DB0576-DB0639 (2 Kbytes) Block Programming/Erasure Enable		R/W
b2	DBWE10	DB0640-DB0703 (2 Kbytes) Block Programming/Erasure Enable		R/W
b3	DBWE11	DB0704-DB0767 (2 Kbytes) Block Programming/Erasure Enable		R/W
b4	DBWE12	DB0768-DB0831 (2 Kbytes) Block Programming/Erasure Enable		R/W
b5	DBWE13	DB0832-DB0895 (2 Kbytes) Block Programming/Erasure Enable		R/W
b6	DBWE14	DB0896-DB0959 (2 Kbytes) Block Programming/Erasure Enable		R/W
b7	DBWE15	DB0960-DB1023 (2 Kbytes) Block Programming/Erasure Enable		R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable rewriting of the DBWEj bit (j = 08 to 15).	R/(W) *1

Note 1. Write data is not retained.

DFLWE1 is a register to enable or disable the DB0512 to DB1023 blocks of the data area (see Figure 46.3) to be programmed or erased. Programming or erasing is enabled or disabled in 2 Kbytes (64 erasure blocks).
Write access to DFLWE1 is valid only when the specific value is written to the upper byte in word access. Data written to the upper byte is not retained.
When on-chip ROM is disabled, the data read from DFLWE1 is 0000h and writing is disabled.

DBWEj Bit (DBj Block Programming/Erasure Enable) (j = 08 to 15)

This bit is used to enable or disable the DB0512 to DB1023 blocks of the data area to be programmed or erased.
The DBWEj bit is used to control programming/erasure of the DBj blocks.
Programming of the DBWEj bit is enabled only in word access when the KEY[7:0] bits are E1h.

46.2.8 Flash P/E Mode Entry Register (FENTRYR)

Address(es): 007F FFB2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FEKEY[7:0]								FENTRYD	—	—	—	FENTRY3	FENTRY2	FENTRY1	FENTRY0
Value after 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	FENTRY0	ROM P/E Mode Entry 0	See section 45, ROM (Flash Memory for Code Storage).	R/W
b1	FENTRY1	ROM P/E Mode Entry 1	See section 45, ROM (Flash Memory for Code Storage).	R/W
b2	FENTRY2	ROM P/E Mode Entry 2	See section 45, ROM (Flash Memory for Code Storage).	R/W
b3	FENTRY3	ROM P/E Mode Entry 3	See section 45, ROM (Flash Memory for Code Storage).	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	E2 DataFlash P/E Mode Entry	0: E2 DataFlash is in read mode 1: E2 DataFlash is in P/E mode	R/W
b15 to b8	FEKEY[7:0]	Key Code	Enable or disable rewriting of the FENTRYD, and FENTRY0 to FENTRY3 bits.	R/(W) *1

Note 1. Write data is not retained.

To place the ROM/E2 DataFlash in ROM P/E mode so that the FCU can accept commands, any of the FENTRYD and FENTRY0 to FENTRY3 bits*1 must be set to 1. Note that if a value is set other than AA01h, AA02h*2, AA04h*2, AA08h*2, and AA80h to FENTRYR, the FSTATR0.ILGLERR bit is set to 1 and the FCU enters the command-locked state.

Write access to FENTRYR is valid only when the specific value is written to the upper byte in word access. Any other writing causes the register to be initialized. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from FENTRYR is 0000h and writing is disabled.

FENTRYR is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

For FSTATR0, see section 45.2.6, Flash Status Register 0 (FSTATR0).

For FRESETR, see section 45.2.11, Flash Reset Register (FRESETR).

Note 1. The FENTRY1, FENTRY2, and FENTRY3 bits are provided when the user areacapacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively.

Note 2. AA02h, AA04h, and AA08h can be written to FENTRYR when the user area capacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively.

FENTRYD Bit (E2 DataFlash P/E Mode Entry)

The FENTRYD bit is used to place the E2 DataFlash in P/E mode.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FRDY bit in FSTATR0 is set to 1
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

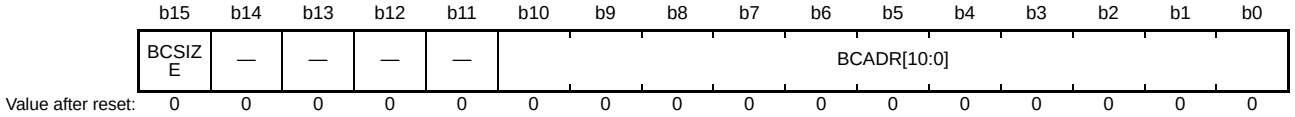
- When the writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRYD bit

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRYD bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

46.2.9 E2 DataFlash Blank Check Control Register (DFLBCCNT)

Address(es): 007F FFCAh



Bit	Symbol	Bit Name	Description	R/W
b10 to b0	BCADR[10:0]	Blank Check Address Setting	Set the address of the area to be checked	R/W
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BCSIZE	Blank Check Size Setting	0: The size of the area to be blank checked is 2 bytes. 1: The size of the area to be blank checked is 2 Kbytes.	R/W

When on-chip ROM is disabled, the data read from DFLBCCNT is 0000h and writing is disabled.

DFLBCCNT is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

For FRESETR, see section 45.2.11, Flash Reset Register (FRESETR).

BCADR[10:0] Bits (Blank Check Address Setting)

These bits are used to set the address of the area to be checked when the size of the area to be checked by a blank check command is 2 bytes (the BCSIZE bit is 0).

When the BCSIZE bit is 0, the start address of the area to be checked is obtained by adding the DFLBCCNT setting value to the read/program/erase enable/disable block start address specified at issuance of a blank check command.

46.2.10 E2 DataFlash Blank Check Status Register (DFLBCSTAT)

Address(es): 007F FFCEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BCST
Value after 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	BCST	Blank Check Status	0: The area to be blank-checked is erased (blank) 1: 0 or 1 is written in the area to be blank-checked	R
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When on-chip ROM is disabled, the data read from DFLBCSTAT is 0000h and writing is disabled.
DFLBCSTAT is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.
For FRESETR, see section 45.2.11, Flash Reset Register (FRESETR).

46.3 Configuration of Memory Area for the E2 DataFlash Memory

The E2 DataFlash memory of products in the RX63N/RX631 Group is configured as a 32-Kbyte data area. The address range occupied by this area is shown in Figure 46.2.
Note that for the data area, the address range for reading is the same as the address range for programming and erasure.

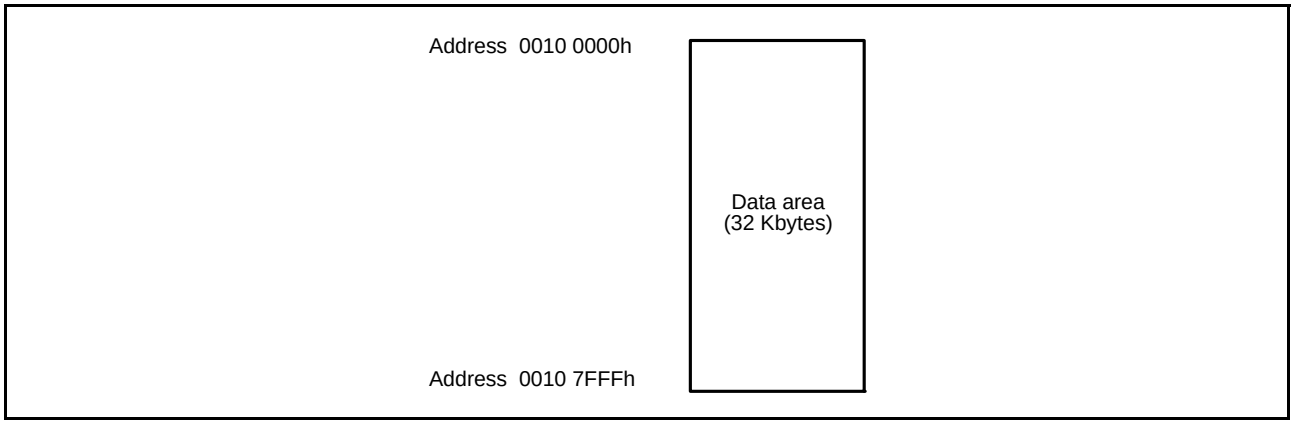


Figure 46.2 Configuration of the Data Area for the E2 DataFlash Memory

46.4 Block Configuration

Figure 46.3 shows how the erasure blocks of the data area are configured. The data area is divided into 1024 blocks of 32 bytes, and erasure proceeds in these block units. Programming proceeds in 2-byte units. For programming in 2-byte units, each unit starts at an address where the value of the one lower-order bit is 0.

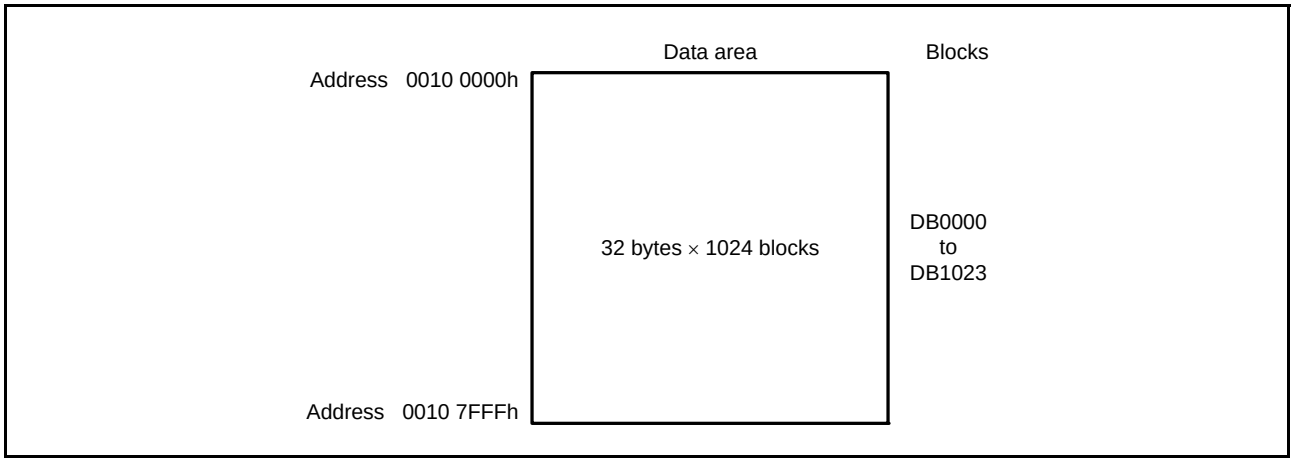


Figure 46.3 Division of the Data Area into Blocks

46.5 Operating Modes Associated with the E2 DataFlash

For the transitions between operating modes, see section 45.5, Operating Modes Associated with the ROM. Reading, programming, and erasing of the data area in an on-board device can proceed if the device is in boot mode, USB boot mode, user boot mode, single-chip mode (with on-chip ROM enabled), or on-chip-ROM-enabled expansion mode.
The differences between modes are indicated in Table 46.3.

Table 46.3 Differences between Modes

Item	Boot Mode	USB Boot Mode/ User Boot Mode	Single-Chip Mode (with On-Chip ROM Enabled) or On-Chip-ROM-Enabled Expansion Mode
Environment for programming and erasure	On-board programming		
Programmable and erasable area	Data area	Data area	Data area
Division into erasure blocks	Possible*1	Possible*1	Possible
Boot program at a reset	Boot program	User boot program	User program

Note 1. All flash memory areas may be erased at the time of booting up. Specified blocks can subsequently be erased. For details, refer to section 45.10.4, ID Code Protection (Boot Mode), section 45.10.2, State Transitions in Boot Mode, and section 45.11.2, State Transitions.

- In boot mode, a host is able to program/erase or read the data area via an SCI.
- In boot mode, on-chip RAM is employed for the embedded program for use in boot mode. For this reason, preserving the contents of on-chip RAM is not possible in this case.
- Booting-up in USB boot mode and user boot mode is from the user boot area. The user boot area of the product as-shipped holds the USB boot program, which is capable of reading from or writing to the user area and data area. Furthermore, rewriting of the user boot area in boot mode can enable reading from or writing to the user area and data area via any desired interface.

46.6 Programming and Erasing the E2 DataFlash Memory

The E2 DataFlash memory is programmed and erased by issuing commands to a dedicated sequencer (FCU) for programming and erasure. The FCU has five modes. For programming and erasure, the mode is changed and then commands for programming and erasure are issued.
The mode transitions required to program or erase the E2 DataFlash and the system of commands are described below. The descriptions apply in common to boot mode, USB boot mode, user boot mode, single-chip mode (with on-chip ROM enabled), and on-chip ROM enabled expansion mode.

46.6.1 FCU Modes

The FCU has five modes or sets of modes. Transitions between modes are caused by writing to the FENTRYR register or issuing FCU commands. Figure 46.4 is a diagram of the FCU mode transitions.

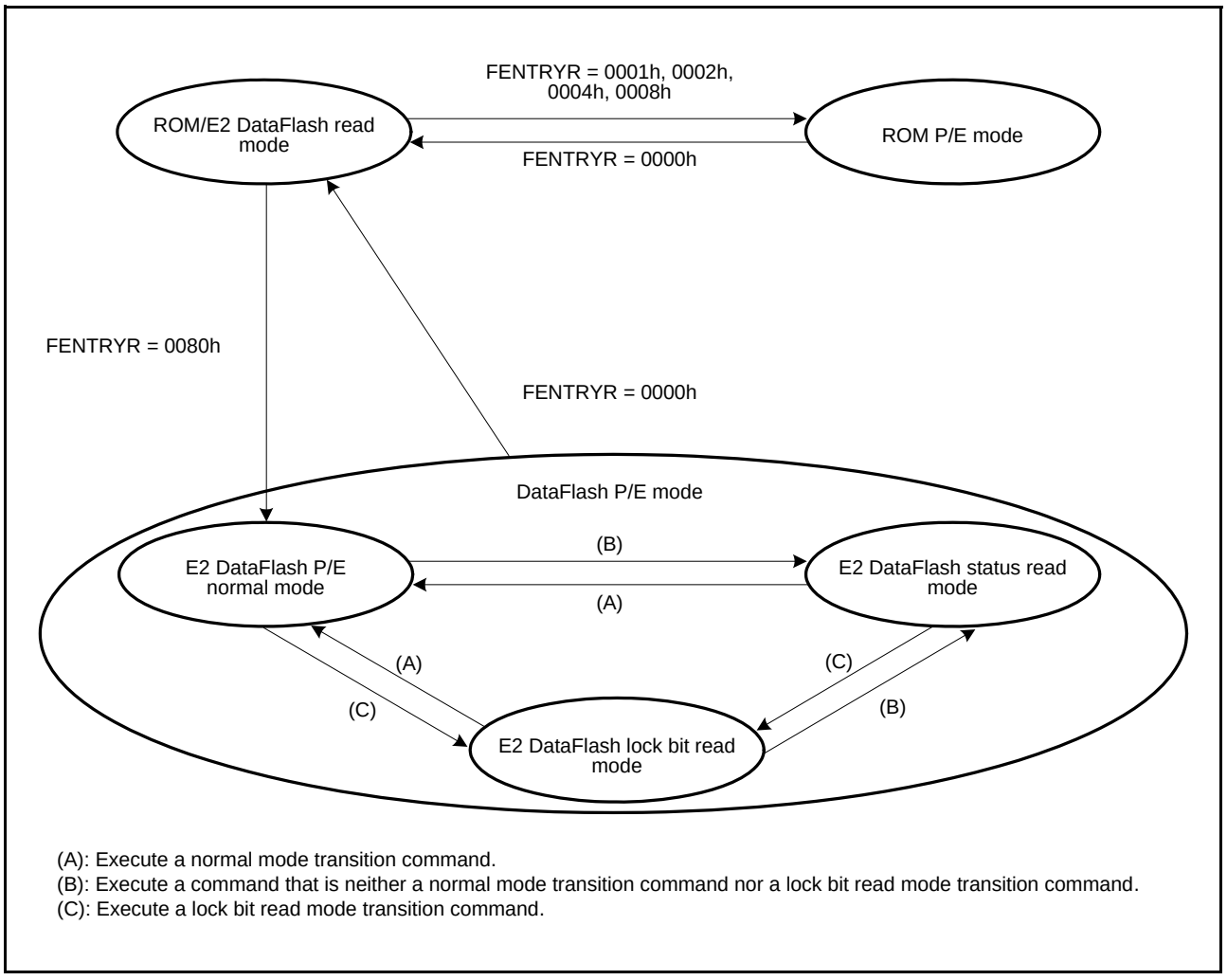


Figure 46.4 Mode Transitions of the FCU (Associated with the E2 DataFlash)

46.6.1.1 ROM P/E Modes

The ROM P/E modes are for programming and erasure of the ROM.
For details on the ROM P/E modes, see section 45.6.1.2, ROM P/E Modes.

46.6.1.2 ROM/E2 DataFlash Read Mode

This mode is for reading the ROM or E2 DataFlash memory. The FCU does not accept commands.
The FCU enters this mode when the FENTRYD and FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits in FENTRYR are all set to 0.

46.6.1.3 E2 DataFlash P/E Modes

These modes are for programming and erasure of the E2 DataFlash memory. Reading out the E2 DataFlash is not possible.
E2 DataFlash P/E normal mode, E2 DataFlash status read mode, and E2 DataFlash lock-bit read mode are the three E2 DataFlash P/E modes.

(1) E2 DataFlash P/E Normal Mode

The transition to E2 DataFlash P/E normal mode is the first transition in the process of programming or erasing the E2 DataFlash. The FCU enters this mode when the FENTRYD bit in FENTRYR is set to 1 and the FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits in FENTRYR are all set to 0 in ROM/E2 DataFlash read mode, or when the normal mode transition command is received in E2 DataFlash P/E modes. Table 46.6 lists the acceptable commands in this mode.

(2) E2 DataFlash Status Read Mode

The E2 DataFlash status read mode is for reading information on the state of the E2 DataFlash.
The FCU enters this mode when a command other than the normal mode transition and lock bit read mode transition command is received in E2 DataFlash P/E modes. E2 DataFlash status read mode encompasses the states where the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred.Table 46.6 lists the acceptable commands in this mode.
Read access to an address within the E2 DataFlash area will actually read the value of the FSTATR0 register. High-speed reading of the ROM is possible.

(3) E2 DataFlash Lock-Bit Read Mode

The E2 DataFlash lock-bit read mode is for reading the values of the lock bits of the E2 DataFlash. However, this is not possible because the E2 DataFlash does not have lock bits.
The FCU enters this mode when a lock-bit read mode transition command is received in E2 DataFlash P/E modes. Table 46.6 lists the acceptable commands in this mode.
Since the E2 DataFlash does not have lock bits, data read in read access to addresses within the E2 DataFlash area are undefined. However, the access does not lead to an E2 DataFlash-access violation. High-speed reading of the ROM is possible.

46.6.2 FCU Commands

FCU commands consist of commands for mode transitions of the FCU and of commands for programming and erasure. Table 46.4 lists the FCU commands available for programming and erasure to the E2 DataFlash.

Table 46.4 FCU Commands for Use with E2 DataFlash Memory

Command	Description
P/E normal mode transition	Changes the mode to normal mode (see section 46.6.3, Connections between FCU Modes and Commands)
Status read mode transition	Changes the mode to status read mode (see section 46.6.3, Connections between FCU Modes and Commands)
Lock bit read mode transition (lock bit read 1)	Changes the mode to lock bit read mode (see section 46.6.3, Connections between FCU Modes and Commands)
Peripheral clock notification	Sets the frequency of the FlashIF clock (FCLK)
Programming	E2 DataFlash programming (in 2-byte units)
Block erasure	E2 DataFlash erasure (in block units)
P/E suspension	Suspends programming/erasure
P/E resumption	Resumes programming/erasure
Status register clearing	Clears the ILGLERR, ERSERR and PRGERR bits in FSTATR0 and releases the FCU from the command-locked state
Lock bit read 2/blank checking	Checks whether the specified block of E2 DataFlash memory has been erased (is blank)

Commands other than the blank-checking command are also for use with the ROM.
The blank-checking command for the E2 DataFlash memory is also used as the lock bit read 2 command for the ROM. That is, when the same command is issued for the ROM, a lock bit of the ROM is read out.
Commands for the FCU are issued by write access to addresses within the E2 DataFlash area.
Table 46.5 lists the formats of the programming commands and the blank checking command. For the formats of FCU commands other than programming and blank checking commands, see section 45.6.2, FCU Commands.
Write access as listed in Table 46.5 and in accord with certain conditions causes the FCU to execute processing for the corresponding command. For details on the conditions for acceptance of the individual FCU commands, see section 46.6.3, Connections between FCU Modes and Commands. For how to use the FCU commands, see section 46.6.4, FCU Command Usage.

Table 46.5 FCU Command Formats (Commands Dedicated to the E2 DataFlash Memory)

Command	Number of Bus Cycles	First Cycle		Second Cycle		Third Cycle		(N+3)th Cycles	
		Address	Data	Address	Data	Address	Data	Address	Data
Programming (2-byte programming; N = 1)	4	EA	E8h	EA	01h	WA	WDn	EA	D0h
Blank checking	2	EA	71h	BA	D0h	—	—	—	—

Address column EA: Address within the E2 DataFlash area. Any address in the range from 0010 0000h to 0010 7FFFh.
WA: Start address of programming data
BA: Block address where reading, programming, or erasure of the E2 DataFlash is enabled or prohibited. Any address in a block where reading, programming, or erasure is enabled or prohibited.
Data columns WDn: nth word of data for programming (n = 1 to N)

46.6.3 Connections between FCU Modes and Commands

The sets of FCU commands that can be accepted in each of the FCU modes are fixed. Furthermore, which commands are acceptable in a given FCU mode varies according to the state of the FCU.
Issuing of an FCU command must follow checking of the FCU's state after transitions of the FCU mode.
Commands that are acceptable in the various FCU modes and states are listed in Table 46.6. Issuing a command that is not currently acceptable leads to the FCU being placed in the command-locked state (see section 46.7.2, Error Protection).
Issuing of an FCU command must follow checking of the values of the acceptable FRDY, ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and of the FCUERR bit in FSTATR1 after transitions of the FCU mode. Furthermore, the CMDLK bit in FASTAT can be checked to see if an error has occurred. The value of the CMDLK bit in FASTAT is the logical OR of the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR bit in FSTATR1.

Table 46.6 Acceptable Commands and the State and Mode (E2 DataFlash P/E Mode) of the FCU

	P/E Normal Mode			**Status Read Mode**									**Lock-Bit Read Mode**		
	Programming Suspended	**Erasure Suspended**	**Other State**	**Programming or Erasure**	**Programming while erasure is suspended**	**Processing to Suspend programming or Erasure**	**Blank Checking**	**Programming Suspended**	**Erasure Suspended**	**Command-locked State** (FRDY = 0)	**Command-locked State** (FRDY = 1)	**Other State**	**Programming Suspended**	**Erasure Suspended**	**Other State**
FSTATR0.FRDY bit	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1
FSTATR0.SUSRDY bit	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
FSTATR0.ERSSPD bit	0	1	0	0	1	0/1	0	0	1	0/1	0/1	0	0	1	0
FSTATR0.PRGSPD bit	1	0	0	0	0	0/1	0	1	0	0/1	0/1	0	1	0	0
FASTAT.CMDLK bit	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P/E Normal mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Status read mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Lock-bit read mode transition (lock bit read 1)	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Peripheral clock notification	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A
Programming	×	*	A	×	×	×	×	×	*	×	×	A	×	*	A
Block erasure	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A
P/E suspension	×	×	×	A	×	×	×	×	×	×	×	×	×	×	×
P/E resumption	A	A	×	×	×	×	×	A	A	×	×	×	A	A	×
Status register clearing	A	A	A	×	×	×	×	A	A	A	A	A	A	A	A
Blank checking	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A

A: Acceptable
*: Only programming is acceptable for blocks other than the block where erasure was suspended
×: Not acceptable

46.6.4 FCU Command Usage

This section shows how to program and erase the E2 DataFlash memory by using programming and block erasure commands, respectively, and how to check the state of erasure of the E2 DataFlash by using the blank check command. For the method for transferring the firmware to the FCU RAM and the ways to use other FCU commands, see section 45.6.4, FCU Command Usage.

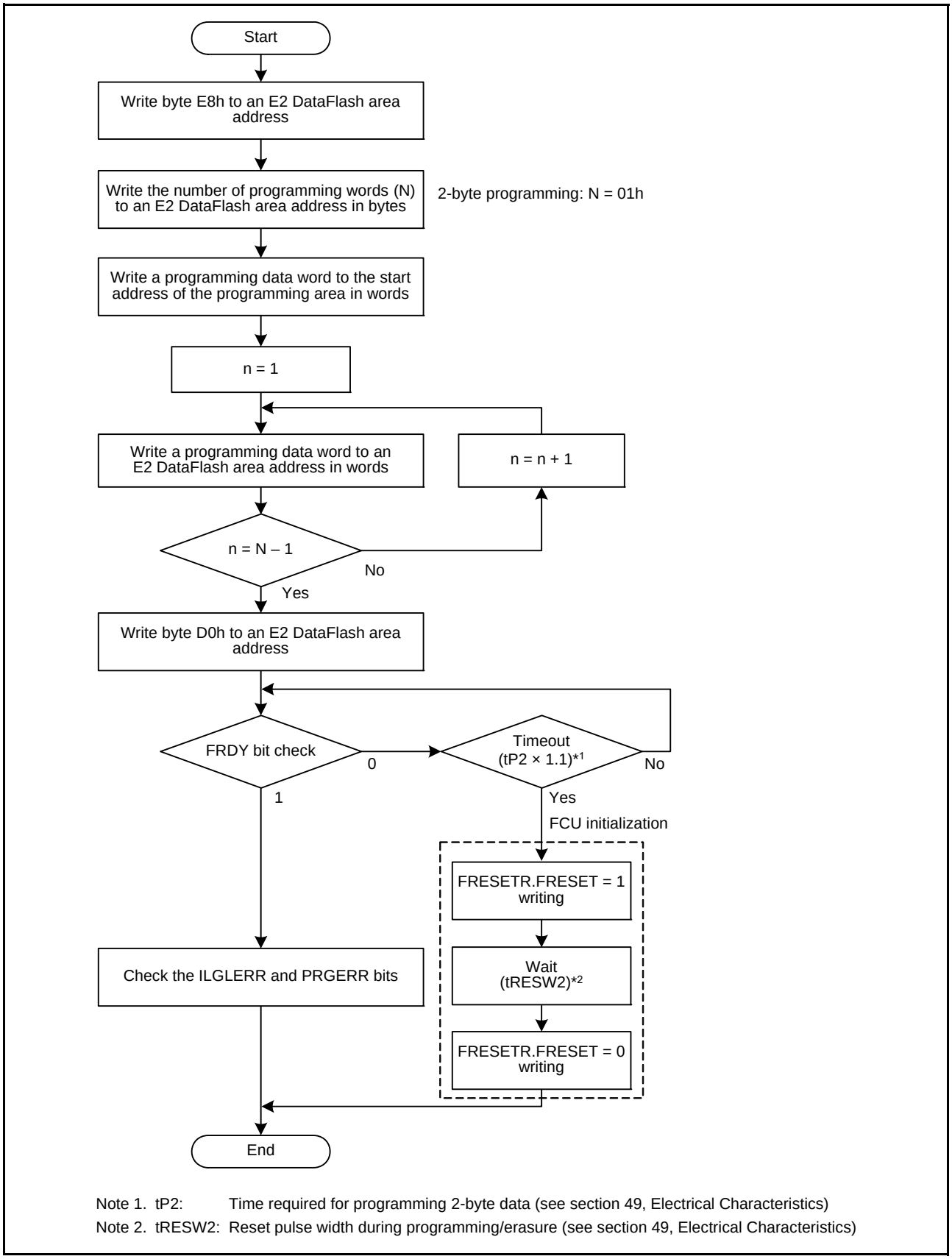
(1) Using the Peripheral Clock Notification Command

This command handles notification of the frequency of the peripheral clock. For details, see section 45.6.4, FCU Command Usage. Set the FENTRYD bit in FENTRYR to 1 and make settings to indicate an address within the E2 DataFlash area.

(2) Programming

To program the E2 DataFlash, use one of the programming commands.
Use byte access to write E8h to an address within the E2 DataFlash area in the first cycle of the programming command, and the number of words (N)*1 to be programmed in the second cycle. Access the peripheral bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the first word of data for programming to the address where the target area for programming starts. This address must be on a 2-byte boundary.
After writing words to addresses in the E2 DataFlash area N times, write byte D0h to an address within the E2 DataFlash area in cycle N + 3; the FCU will then start actual programming of the E2 DataFlash. Read the FRDY bit in FSTATR0 to confirm the completion of E2 DataFlash programming.
If the area accessed in the third cycle to cycle N + 2 includes addresses that do not require programming, write FFFFh as the programming data for those addresses. To ignore the programming and erasure protection provided by the DFLWEy (y = 0, 1), set the program/erase enable bit for the target block to 1 before programming starts.
Figure 46.5 shows the procedure for E2 DataFlash programming.

Note 1. N = 01h for 2-byte programming.



Note 1. tP2: Time required for programming 2-byte data (see section 49, Electrical Characteristics)
Note 2. tRESW2: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

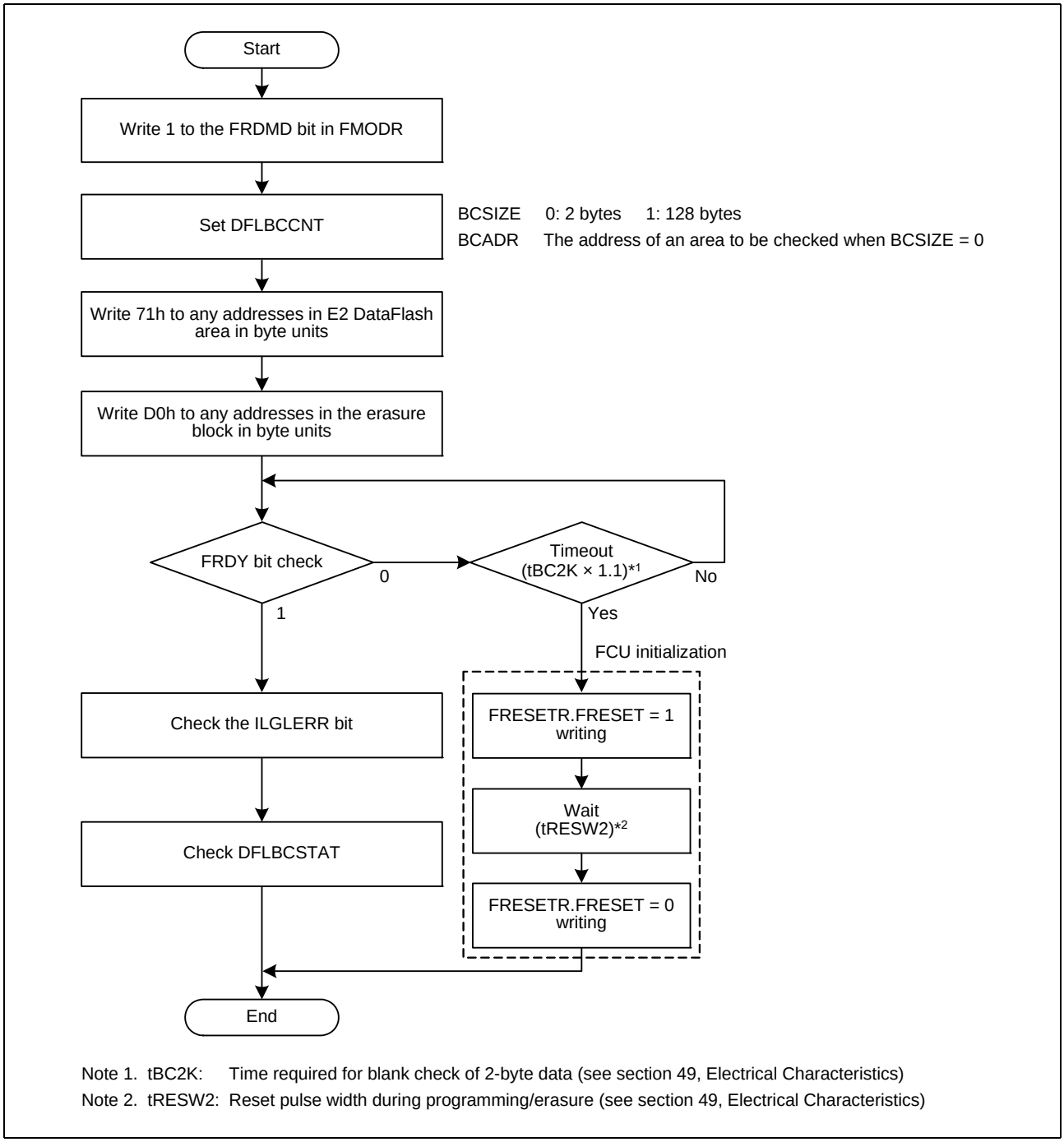
Figure 46.5 Procedure for E2 DataFlash Programming

(3) Erasure

To erase the E2 DataFlash, use the block erasure command. The E2 DataFlash is erased in the same way as the ROM (see section 45, ROM (Flash Memory for Code Storage)).
Note that the E2 DataFlash has a programming and erasure protection function that is controlled by DFLWEy (y = 0, 1). Erasure can only be performed with protection provided by the DFLWEy setting disabled, so set the programming/erasure enable bit for the target erasure block to 1 before issuing the erasure command.

(4) Blank Checking

Since using the CPU to read erased areas of the E2 DataFlash produces undefined values, the blank checking command should be used to check whether the E2 DataFlash has actually been erased. To make the blank checking command available for use, start by setting the FRDMD bit in FMODR to 1 to enable the command, and then specify the size and start address of the target area in DFLBCCNT. When the DFLBCCNT.BCSIZE bit is 1, checking can be performed on the entire area (2 Kbytes) as specified in the second cycle of the blank check command. When the BCSIZE bit is 0, checking can be performed on the 2-byte range starting from the address obtained by adding the start address of the area as specified in the second cycle of the command and the value held by DFLBCCNT. In the first cycle of the command sequence, the value 71h is written as a byte unit to an address in the E2 DataFlash. In the second cycle, when the value D0h is written as a byte unit to an address in the erasure block within the target area, the FCU starts blank checking of the E2 DataFlash. Test the FRDY bit in the FSTATR0 register to check whether or not the check is complete. On completion of blank checking, check the BCST bit of DFLBCSTAT to see whether the target area has been erased or is filled with 0s and/or 1s.
Figure 46.6 shows the procedure for blank checking of the E2 DataFlash.



Note 1. tBC2K: Time required for blank check of 2-byte data (see section 49, Electrical Characteristics)
Note 2. tRESW2: Reset pulse width during programming/erasure (see section 49, Electrical Characteristics)

Figure 46.6 Procedure for Blank Checking of the E2 DataFlash

46.7 Protection

There are two types of E2 DataFlash programming/erasure protection: software protection and error protection.

46.7.1 Software Protection

In the software protection function, control register settings are used to disable E2 DataFlash programming, erasure, and reading. If an attempt is made to issue a programming, an erasure, or a reading command for the E2 DataFlash and the command violates current software protection, the FCU detects the error and enters the command-locked state.

(1) Protection through FWEPROR

If the FLWE[1:0] bits in FWEPROR are not set to 01b, programming cannot be performed in any mode.

(2) Protection through FENTRYR

When the FENTRYD bit in FENTRYR is 0, the ROM/E2 DataFlash read mode is selected. Since the FCU does not accept commands in ROM/E2 DataFlash read mode, E2 DataFlash programming and erasure are disabled. If an attempt is made to issue an FCU command for the E2 DataFlash in ROM/E2 DataFlash read mode, the FCU detects an illegal command error and enters the command-locked state (see section 46.7.2, Error Protection).

(3) Protection through DFLWEy

When the DBWEj (j = 00 to 15) bit in DFLWEy (y = 0, 1) is 0, programming and erasure of block DBj in the data area is disabled. If an attempt is made to program or erase block DBj while the DBWEj bit is 0, the FCU detects a programming/erasure protection error and enters the command-locked state (see section 46.7.2, Error Protection).

(4) Protection through DFLREy

When the DBREj (j = 00 to 15) bit in DFLREy (y = 0, 1) is 0, reading of block DBj in the data area is disabled. If an attempt is made to read block DBj while the DBREj bit is 0, the FCU detects a read protection error and enters the command-locked state (see section 46.7.2, Error Protection).

46.7.2 Error Protection

Error protection is the detection of errors in the issuing of FCU commands and of prohibited access, and response in the form of notification of the FCU malfunction and prohibition of the reception of further commands by the FCU (the FCU enters the command-locked state). When the FCU enters the command-locked state (FASTAT.CMDLK bit is 1), one or several of the status bits (FSTATR0.ILGLERR, ERSERR, and PRGERR bits, FSTATR1.FCUERR bit, and FASTAT.DFLAE, DFLRPE, and DFLWPE bits) are set to 1 and programming and erasure of the E2 DataFlash are prohibited. To release the FCU from the command-locked state, a status register clearing command must be issued with FASTAT set to 10h.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFERR) interrupt will be generated if the FCU enters the command-locked state (the CMDLK bit in FASTAT becomes 1). While an E2 DataFlash-related interrupt enable bit (DFLAEIE, DFLRPEIE, or DFLWPEIE) in FAEINT is 1, an FIFERR interrupt will also be generated if the corresponding status bit (DFLAE, DFLRPE, or DFLWPE) in FASTAT becomes 1.

Table 46.7 shows the error protection types for the E2 DataFlash and the values of the status bits (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the DFLAE, DFLRPE, and DFLWPE bits in FASTST) after the detection of each type of error. For the error protection types used in common by the ROM and E2 DataFlash (FENTRYR setting error, most illegal command errors, erasing errors, programming errors, and FCU errors, see section 45.8.2, Error Protection.

If the FCU enters the command-locked state due to a command other than a suspension command issued during programming or erasure processing, the FCU continues programming or erasing the E2 DataFlash. In this state, the P/E suspension command cannot suspend programming or erasure. If a command is issued in the command-locked state, the ILGLERR bit becomes 1.

Table 46.7 Error Protection Types (for E2 DataFlash Only)

Error	Description	ILGLERR	ERSERR	PRGERR	DFLAE	DFLRPE	DFLWPE
Illegal command error	The value specified in the second cycle of a programming command was other than 01h.	1	0	0	0	0	0
	A lock bit programming command was issued for an area in the E2 DataFlash while the FENTRYD bit of FENTRYR register was set to 1.	1	0	0	0	0	0
E2 DataFlash access error	A read access command was issued for the E2 DataFlash area while FENTRYD = 1 in FENTRYR in E2 DataFlash P/E normal mode.	1	0	0	1	0	0
	A write access command was issued for the E2 DataFlash area while FENTRYD = 0.	1	0	0	1	0	0
	A write access command was issued for the E2 DataFlash area while any of bits FENTRY3, FENTRY2, FENTRY1, and FENTRY0 in FENTRYR was 1.	1	0	0	1	0	0
E2 DataFlash read protect error	A read access command was issued for the E2 DataFlash area while it was protected against reading by the DFLREy (y = 0, 1) setting.	1	0	0	0	1	0
E2 DataFlash programming protect error	A program/block erase command was issued for the E2 DataFlash area while it was protected against programming and erasure by the DFLWEy (y = 0, 1) setting.	1	0	0	0	0	1

46.8 Boot Mode

To program or erase the data area in boot mode, send control commands and programming data from the host. For the system configuration and settings in boot mode, see section 46.8, Boot Mode. This section describes only the commands dedicated for the E2 DataFlash.

46.8.1 Inquiry/Selection Host Commands

Table 46.8 lists the inquiry/selection host commands dedicated to the E2 DataFlash. The data area inquiry and data area information inquiry commands are used in the step of “Inquiry regarding area programming information” in the flowchart shown in Figure 45.28, Example of Procedure to Use Inquiry/Selection Host Commands for User Area/User Boot Area in section 45.10.6, Inquiry/Selection Host Command Wait State.

Table 46.8 Inquiry/Selection Host Commands (only for E2 DataFlash)

Host Command Name	Function
Data area inquiry	Inquires regarding the availability of data area
Data area information inquiry	Inquires regarding the number of data areas and the start and end addresses

Each host command is described in detail below. The “command” in the description indicates a command sent from the host to the RX63N/RX631 and the “response” indicates a response sent from the RX63N/RX631 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX63N/RX631 becomes 00h.

(1) Data Area Inquiry

In response to a data area inquiry command sent from the host, the RX63N/RX631 returns the information concerning the availability of data areas.

Command	2Ah			
Response	3Ah	Size	Area availability	SUM

Size (1 byte): Number of characters in the area availability field (fixed at 1)

Area availability (1 byte): Availability of data areas (fixed at 21h)
21h: Data area is available

SUM (1 byte): Checksum

(2) Data Area Information Inquiry

In response to a data area information inquiry command sent from the host, the RX63N/RX631 returns the number of data area counts and their addresses.

Command	2Bh		
Response	3Bh	Size	Area count
	Area start address		
	Area end address		
	Area start address		
	Area end address		
	:		
	Area start address		
	Area end address		
	SUM		

Size (1 byte):	Total number of bytes in the area count, area start address, and area end address fields
Area count (1 byte):	Number of data area counts (consecutive areas are counted as one area)
Area start address (4 bytes):	Start address of a data area
Area end address (4 bytes):	End address of a data area
SUM (1 byte):	Checksum

The information concerning the block configuration in the data area is included in the response to the erasure block information inquiry command (see section 45.10.6, Inquiry/Selection Host Command Wait State).

46.8.2 Programming/Erasing Host Commands

Table 46.9 lists the programming/erasing host commands dedicated to the E2 DataFlash. E2 DataFlash-dedicated host commands are provided only for checksum and blank check of the data area; the programming, erasing, and reading commands are used in common for the ROM and E2 DataFlash.

To program the data area, issue from the host a user area programming selection command and then a 256-byte programming command specifying a data area address as the programming address. To erase the data area, issue an erasure selection command and then a block erasure command specifying an erasure block in the data area. The information concerning the erasure block in the data area is included in the response to the erasure block information inquiry command. To read data from the data area, select the user area through a memory read command specifying a data area address as the read address.

For the user area programming selection, user boot area programming selection, 256-byte programming, erasure selection, block erasure, and memory read commands, refer to section 45.10.8, Programming/Erasure Host Command Wait State. For the erasure block information inquiry command, refer to section 45.10.6, Inquiry/Selection Host Command Wait State.

Table 46.9 Programming/Erasure Host Commands (only for E2 DataFlash)

Host Command Name	Function
Data area checksum	Performs checksum verification for the data area
Data area blank check	Checks whether the data area is blank

Each host command is described in detail below. The “command” in the description indicates a command sent from the host to the RX63N/RX631 and the “response” indicates a response sent from the RX63N/RX631 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX63N/RX631 becomes 00h.

(1) Data Area Checksum

In response to a data area checksum command sent from the host, the RX63N/RX631 sums the data area data in byte units and returns the result (checksum).

Command	61h			
Response	71h	Size	Area checksum	SUM

Size (1 byte):	Number of bytes in the area checksum field (fixed at 4)
Area checksum (4 bytes):	Checksum of the data area data
SUM (4 bytes):	Checksum (for the response data)

(2) Data Area Blank Check

In response to a data area blank check command sent from the host, the RX63N/RX631 checks whether the data area is completely erased. When the data area is completely erased, the RX63N/RX631 returns a response (06h). If the data area has an unerased area, the RX63N/RX631 returns an error response (sends E2h and 52h in that order).

Command	62h	
Response	06h	
Error response	E2h	52h

46.9 Usage Notes

(1) Protection of Data Area Immediately after a Reset

As the initial values of DFLREy and DFLWEy (y = 0, 1) are 0000h, programming, erasure, and reading of the data area are disabled immediately after a reset. To read data from the data area, set DFLREy appropriately before accessing the data area. To program or erase the data area, set DFLWEy appropriately before issuing an FCU command for programming or erasure. If an attempt is made to read, program, or erase the data area without setting the registers, the FCU detects the error and enters the command-locked state.

(2) Reading in Low-Speed Operating Mode 2

The E2 DataFlash memory is not readable in low-speed operating mode 2. Place the chip in another mode if reading is to proceed.

(3) Other Points to Note

The other points to note are the same as for the ROM. See section 45.14, Usage Notes of section 45, ROM (Flash Memory for Code Storage).
However, for the E2 DataFlash, notes on programming/erasing also apply to the blank checking.

47. Boundary Scan

The RX63N/RX631 Group 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).

47.1 Features

Table 47.1 lists the specifications of boundary scan.
Figure 47.1 shows a block diagram of the boundary scan function.

Table 47.1 Specifications of Boundary Scan

Item	Description
Boundary scan enabled/disabled	Boundary can is enabled when the EMLE pin is driven low and the BSCANP pin is driven high.
Dedicated boundary scan pins	The following pins are dedicated for 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

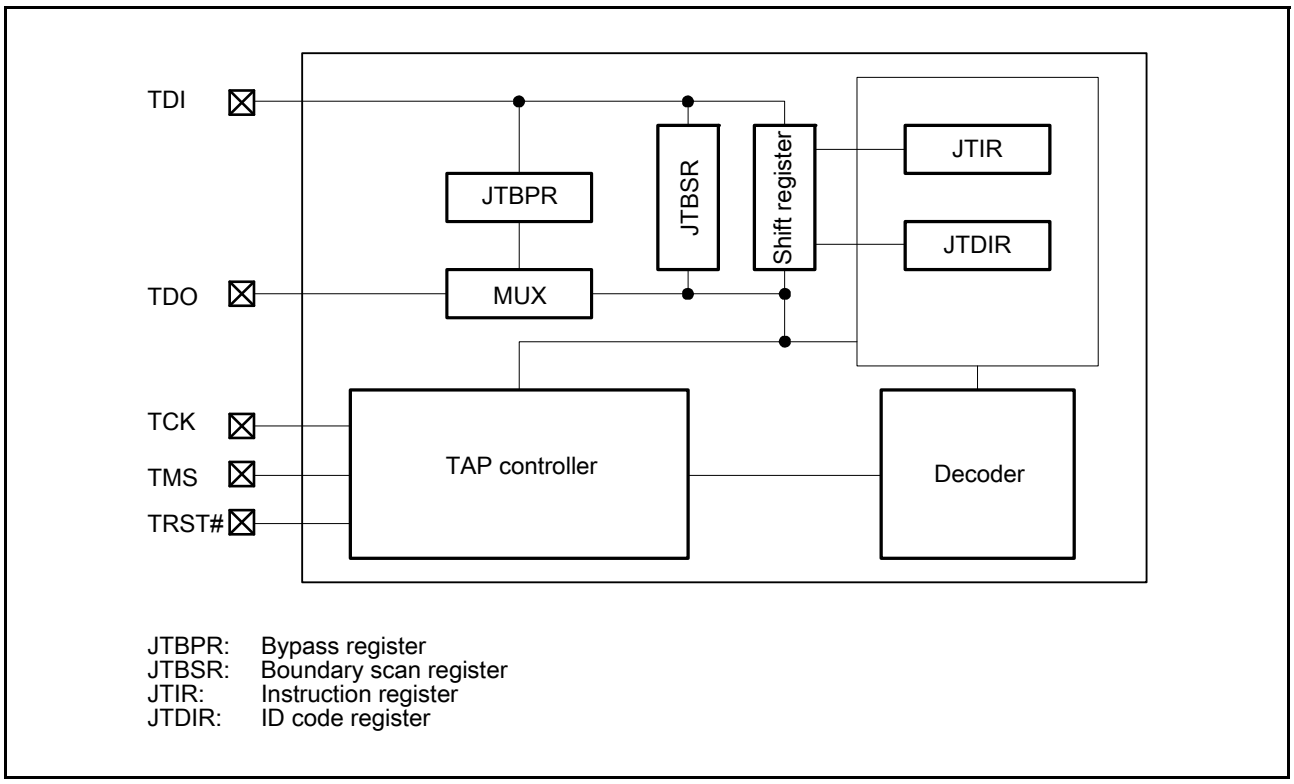


Figure 47.1 Block Diagram of JTAG

Table 47.2 shows the I/O pins used in the boundary scan function.

Table 47.2 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

47.2 Register Descriptions

Table 47.3 lists the boundary scan registers.

Table 47.3 List of Boundary Scan Registers

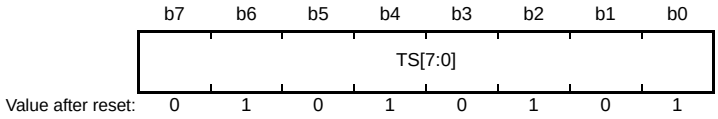
Register Name	Symbol	Value after Reset	Address	Access Size
Instruction register	JTIR	55h	—	—
ID code register	JTIDR	080B B447h	—	—
Bypass register	JTBPR	undefined	—	—
Boundary scan register	JTBSR	undefined	—	—

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 47.6, 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 47.4 shows the availability of serial transfer for the registers.

Table 47.4 Serial Transfer for the Registers

Register Name	Serial Input	Serial Output
Instruction register (JTIR)	Available	Not available
ID code register (JTIDR)	Not available	Available
Bypass register (JTBPR)	Available	Available
Boundary scan register (JTBSR)	Available	Available

47.2.1 Instruction Register (JTIR)



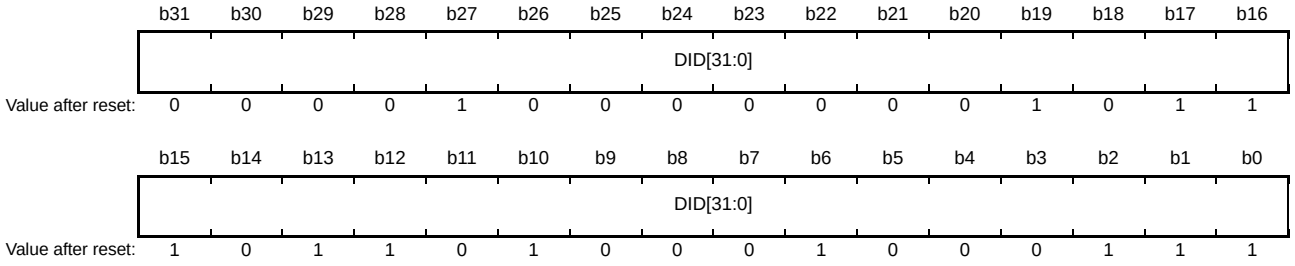
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TS[7:0]	Test Bit Set	The command configuration is as shown in Table 47.5.	—

Table 47.5 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.

47.2.2 ID Code Register (JTID)



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.	—

JTID data is output from the TDO pin when the IDCODE instruction has been executed.

47.2.3 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.

47.2.4 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 47.6 shows the correspondence between the JTBSR bits and the pins of this LSI.
The value after reset is undefined.

Table 47.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (1/9)

From TDI			
Pin No.	**Pin Name**	**Input/Output**	**Bit Name**
B1	P05	Output	372
		Output enable	371
		Input	370
D3	P03	Output	369
		Output enable	368
		Input	367
D2	P02	Output	366
		Output enable	365
		Input	364
D1	P01	Output	363
		Output enable	362
		Input	361
D4	P00	Output	360
		Output enable	359
		Input	358
E3	PF5	Output	357
		Output enable	356
		Input	355
E1	PJ5	Output	354
		Output enable	353
		Input	352
F3	PJ3	Output	351
		Output enable	350
		Input	349
H4	P35	Input	348
K1	P33	Output	347
		Output enable	346
		Input	345
K2	P32	Output	344
		Output enable	343
		Input	342
L4	P25	Output	341
		Output enable	340
		Input	339
M3	P24	Output	338
		Output enable	337
		Input	336
N2	P23	Output	335
		Output enable	334
		Input	333
N3	P22	Output	332
		Output enable	331
		Input	330

Table 47.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (2/9)

Pin No.	**Pin Name**	**Input/Output**	**Bit Name**
R1	P21	Output	329
		Output enable	328
		Input	327
R2	P20	Output	320
		Output enable	319
		Input	318
P2	P17	Output	317
		Output enable	316
		Input	315
P3	P87	Output	314
		Output enable	313
		Input	312
R3	P16	Output	311
		Output enable	310
		Input	309
M4	P86	Output	308
		Output enable	307
		Input	306
N4	P15	Output	305
		Output enable	304
		Input	303
P4	P14	Output	302
		Output enable	301
		Input	300
R4	P85	Output	299
		Output enable	298
		Input	297
M5	P13	Output	296
		Output enable	295
		Input	294
N5	P12	Output	293
		Output enable	292
		Input	291
R5	P11	Output	290
		Output enable	289
		Input	288
P5	P10	Output	287
		Output enable	286
		Input	285
N6	P57	Output	284
		Output enable	283
		Input	282

Table 47.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (3/9)

Pin No.	Pin Name	Input/Output	Bit Name
M6	P56	Output	281
		Output enable	280
		Input	279
M6	P55	Output	278
		Output enable	277
		Input	276
M7	P54	Output	275
		Output enable	274
		Input	273
M8	P53	Output	272
		Output enable	271
		Input	270
R9	P84	Output	269
		Output enable	268
		Input	267
P9	P52	Output	266
		Output enable	265
		Input	264
N9	P51	Output	263
		Output enable	262
		Input	261
M9	P50	Output	260
		Output enable	259
		Input	258
P10	P83	Output	257
		Output enable	256
		Input	255
N10	PC7	Output	254
		Output enable	253
		Input	252
P11	PC6	Output	251
		Output enable	250
		Input	249
M10	PC5	Output	248
		Output enable	247
		Input	246
N11	P82	Output	245
		Output enable	244
		Input	243
M11	P81	Output	242
		Output enable	241
		Input	240
R12	P80	Output	239
		Output enable	238
		Input	237

Table 47.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (4/9)

Pin No.	Pin Name	Input/Output	Bit Name
P12	PC4	Output	236
		Output enable	235
		Input	234
N12	PC3	Output	233
		Output enable	232
		Input	231
M12	P77	Output	230
		Output enable	229
		Input	228
R13	P76	Output	227
		Output enable	226
		Input	225
P13	PC2	Output	224
		Output enable	223
		Input	222
P14	P75	Output	221
		Output enable	220
		Input	219
R14	P74	Output	218
		Output enable	217
		Input	216
R15	PC1	Output	215
		Output enable	214
		Input	213
N13	PC0	Output	212
		Output enable	211
		Input	210
N14	P73	Output	209
		Output enable	208
		Input	207
M13	PB7	Output	206
		Output enable	205
		Input	204
L12	PB6	Output	203
		Output enable	202
		Input	201
M14	PB5	Output	200
		Output enable	199
		Input	198
M15	PB4	Output	197
		Output enable	196
		Input	195
L13	PB3	Output	194
		Output enable	193
		Input	192

Table 47.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (5/9)

Pin No.	Pin Name	Input/Output	Bit Name
K12	PB2	Output	191
		Output enable	190
		Input	189
L14	PB1	Output	188
		Output enable	187
		Input	186
L15	P72	Output	185
		Output enable	184
		Input	183
K13	P71	Output	182
		Output enable	181
		Input	180
K15	PB0	Output	179
		Output enable	178
		Input	177
J14	PA7	Output	176
		Output enable	175
		Input	174
J15	PA6	Output	173
		Output enable	172
		Input	171
J12	PA5	Output	170
		Output enable	169
		Input	168
H12	PA4	Output	167
		Output enable	166
		Input	165
H13	PA3	Output	164
		Output enable	163
		Input	162
H15	PG7	Output	161
		Output enable	160
		Input	159
H14	PA2	Output	158
		Output enable	157
		Input	156
G13	PG6	Output	155
		Output enable	154
		Input	153
G14	PA1	Output	152
		Output enable	151
		Input	150
G12	PG5	Output	149
		Output enable	148
		Input	147

Table 47.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (6/9)

Pin No.	Pin Name	Input/Output	Bit Name
F14	PA0	Output	146
		Output enable	145
		Input	144
F13	PG4	Output	143
		Output enable	142
		Input	141
E15	P67	Output	140
		Output enable	139
		Input	138
E14	PG3	Output	137
		Output enable	136
		Input	135
F12	P66	Output	134
		Output enable	133
		Input	132
E13	PG2	Output	131
		Output enable	130
		Input	129
D15	P65	Output	128
		Output enable	127
		Input	126
D14	PE7	Output	125
		Output enable	124
		Input	123
E12	PE6	Output	122
		Output enable	121
		Input	120
C15	P70	Output	119
		Output enable	118
		Input	117
D12	PE5	Output	116
		Output enable	115
		Input	114
C13	PE4	Output	113
		Output enable	112
		Input	111
B15	PE3	Output	110
		Output enable	109
		Input	108
A15	PE2	Output	107
		Output enable	106
		Input	105
A14	PE1	Output	104
		Output enable	103
		Input	102

Table 47.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (7/9)

Pin No.	Pin Name	Input/Output	Bit Name
B14	PE0	Output	101
		Output enable	100
		Input	99
B13	P64	Output	98
		Output enable	97
		Input	96
A13	P63	Output	95
		Output enable	94
		Input	93
C12	P62	Output	92
		Output enable	91
		Input	90
D11	P61	Output	89
		Output enable	88
		Input	87
A12	P60	Output	86
		Output enable	85
		Input	84
D10	PD7	Output	83
		Output enable	82
		Input	81
B11	PG1	Output	80
		Output enable	79
		Input	78
A11	PD6	Output	77
		Output enable	76
		Input	75
C10	PG0	Output	74
		Output enable	73
		Input	72
D9	PD5	Output	71
		Output enable	70
		Input	69
B10	PD4	Output	68
		Output enable	67
		Input	66
A10	P97	Output	65
		Output enable	64
		Input	63
C9	PD3	Output	62
		Output enable	61
		Input	60
B9	P96	Output	59
		Output enable	58
		Input	57

Table 47.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (8/9)

Pin No.	Pin Name	Input/Output	Bit Name
C8	PD2	Output	56
		Output enable	55
		Input	54
D7	P95	Output	53
		Output enable	52
		Input	51
B8	PD1	Output	50
		Output enable	49
		Input	48
A8	P94	Output	47
		Output enable	46
		Input	45
C7	PD0	Output	44
		Output enable	43
		Input	42
D6	P93	Output	41
		Output enable	40
		Input	39
B7	P92	Output	38
		Output enable	37
		Input	36
B6	P91	Output	35
		Output enable	34
		Input	33
C6	P90	Output	32
		Output enable	31
		Input	30
B5	P47	Output	29
		Output enable	28
		Input	27
A5	P46	Output	26
		Output enable	25
		Input	24
C5	P45	Output	23
		Output enable	22
		Input	21
D5	P44	Output	20
		Output enable	19
		Input	18
C4	P43	Output	17
		Output enable	16
		Input	15
A4	P42	Output	14
		Output enable	13
		Input	12

Table 47.6 Boundary Scan Register
177-Pin TFLGA/176-Pin LFBGA (9/9)

Pin No.	Pin Name	Input/Output	Bit Name
B4	P41	Output	11
		Output enable	10
		Input	9
B3	P40	Output	8
		Output enable	7
		Input	6
B2	P07	Output	5
		Output enable	4
		Input	3
		To TDO	

Table 47.7 Boundary Scan Register (145-Pin TFLGA) (1/8)

From TDI			
Pin No.	**Pin Name**	**Input/Output**	**Bit Name**
B1	P05	Output	372
		Output enable	371
		Input	370
D3	P03	Output	369
		Output enable	368
		Input	367
D2	P02	Output	366
		Output enable	365
		Input	364
D1	P01	Output	363
		Output enable	362
		Input	361
D4	P00	Output	360
		Output enable	359
		Input	358
E3	PF5	Output	357
		Output enable	356
		Input	355
E1	PJ5	Output	354
		Output enable	353
		Input	352
F3	PJ3	Output	351
		Output enable	350
		Input	349
H4	P35	Output	348
K1	P33	Output	347
		Output enable	346
		Input	345
K2	P32	Output	344
		Output enable	343
		Input	342
L4	P25	Output	341
		Output enable	340
		Input	339
M3	P24	Output	338
		Output enable	337
		Input	336
N2	P23	Output	335
		Output enable	334
		Input	333
N3	P22	Output	332
		Output enable	331
		Input	330

Table 47.7 Boundary Scan Register (145-Pin TFLGA) (2/8)

Pin No.	Pin Name	Input/Output	Bit Name
R1	P21	Output	329
		Output enable	328
		Input	327
R2	P20	Output	320
		Output enable	319
		Input	318
P2	P17	Output	317
		Output enable	316
		Input	315
P3	P87	Output	314
		Output enable	313
		Input	312
R3	P16	Output	311
		Output enable	310
		Input	309
M4	P86	Output	308
		Output enable	307
		Input	306
N4	P15	Output	305
		Output enable	304
		Input	303
P4	P14	Output	302
		Output enable	301
		Input	300
M5	P13	Output	296
		Output enable	295
		Input	294
N5	P12	Output	293
		Output enable	292
		Input	291
M6	P56	Output	281
		Output enable	280
		Input	279
M6	P55	Output	278
		Output enable	277
		Input	276
M7	P54	Output	275
		Output enable	274
		Input	273
M8	P53	Output	272
		Output enable	271
		Input	270

Table 47.7 Boundary Scan Register (145-Pin TFLGA) (3/8)

Pin No.	Pin Name	Input/Output	Bit Name
P9	P52	Output	266
		Output enable	265
		Input	264
N9	P51	Output	263
		Output enable	262
		Input	261
M9	P50	Output	260
		Output enable	259
		Input	258
P10	P83	Output	257
		Output enable	256
		Input	255
N10	PC7	Output	254
		Output enable	253
		Input	252
P11	PC6	Output	251
		Output enable	250
		Input	249
M10	PC5	Output	248
		Output enable	247
		Input	246
N11	P82	Output	245
		Output enable	244
		Input	243
M11	P81	Output	242
		Output enable	241
		Input	240
R12	P80	Output	239
		Output enable	238
		Input	237
P12	PC4	Output	236
		Output enable	235
		Input	234
N12	PC3	Output	233
		Output enable	232
		Input	231
M12	P77	Output	230
		Output enable	229
		Input	228
R13	P76	Output	227
		Output enable	226
		Input	225
P13	PC2	Output	224
		Output enable	223
		Input	222

Table 47.7 Boundary Scan Register (145-Pin TFLGA) (4/8)

Pin No.	Pin Name	Input/Output	Bit Name
P14	P75	Output	221
		Output enable	220
		Input	219
R14	P74	Output	218
		Output enable	217
		Input	216
R15	PC1	Output	215
		Output enable	214
		Input	213
N13	PC0	Output	212
		Output enable	211
		Input	210
N14	P73	Output	209
		Output enable	208
		Input	207
M13	PB7	Output	206
		Output enable	205
		Input	204
L12	PB6	Output	203
		Output enable	202
		Input	201
M14	PB5	Output	200
		Output enable	199
		Input	198
M15	PB4	Output	197
		Output enable	196
		Input	195
L13	PB3	Output	194
		Output enable	193
		Input	192
K12	PB2	Output	191
		Output enable	190
		Input	189
L14	PB1	Output	188
		Output enable	187
		Input	186
L15	P72	Output	185
		Output enable	184
		Input	183
K13	P71	Output	182
		Output enable	181
		Input	180
K15	PB0	Output	179
		Output enable	178
		Input	177

Table 47.7 Boundary Scan Register (145-Pin TFLGA) (5/8)

Pin No.	Pin Name	Input/Output	Bit Name
J14	PA7	Output	176
		Output enable	175
		Input	174
J15	PA6	Output	173
		Output enable	172
		Input	171
J12	PA5	Output	170
		Output enable	169
		Input	168
H12	PA4	Output	167
		Output enable	166
		Input	165
H13	PA3	Output	164
		Output enable	163
		Input	162
H14	PA2	Output	158
		Output enable	157
		Input	156
G14	PA1	Output	152
		Output enable	151
		Input	150
F14	PA0	Output	146
		Output enable	145
		Input	144
E15	P67	Output	140
		Output enable	139
		Input	138
F12	P66	Output	134
		Output enable	133
		Input	132
D15	P65	Output	128
		Output enable	127
		Input	126
D14	PE7	Output	125
		Output enable	124
		Input	123
E12	PE6	Output	122
		Output enable	121
		Input	120
C15	P70	Output	119
		Output enable	118
		Input	117
D12	PE5	Output	116
		Output enable	115
		Input	114

Table 47.7 Boundary Scan Register (145-Pin TFLGA) (6/8)

Pin No.	Pin Name	Input/Output	Bit Name
C13	PE4	Output	113
		Output enable	112
		Input	111
B15	PE3	Output	110
		Output enable	109
		Input	108
A15	PE2	Output	107
		Output enable	106
		Input	105
A14	PE1	Output	104
		Output enable	103
		Input	102
B14	PE0	Output	101
		Output enable	100
		Input	99
B13	P64	Output	98
		Output enable	97
		Input	96
A13	P63	Output	95
		Output enable	94
		Input	93
C12	P62	Output	92
		Output enable	91
		Input	90
D11	P61	Output	89
		Output enable	88
		Input	87
A12	P60	Output	86
		Output enable	85
		Input	84
D10	PD7	Output	83
		Output enable	82
		Input	81
A11	PD6	Output	77
		Output enable	76
		Input	75
D9	PD5	Output	71
		Output enable	70
		Input	69
B10	PD4	Output	68
		Output enable	67
		Input	66
C9	PD3	Output	62
		Output enable	61
		Input	60

Table 47.7 Boundary Scan Register (145-Pin TFLGA) (7/8)

Pin No.	Pin Name	Input/Output	Bit Name
C8	PD2	Output	56
		Output enable	55
		Input	54
B8	PD1	Output	50
		Output enable	49
		Input	48
C7	PD0	Output	44
		Output enable	43
		Input	42
D6	P93	Output	41
		Output enable	40
		Input	39
B7	P92	Output	38
		Output enable	37
		Input	36
B6	P91	Output	35
		Output enable	34
		Input	33
C6	P90	Output	32
		Output enable	31
		Input	30
B5	P47	Output	29
		Output enable	28
		Input	27
A5	P46	Output	26
		Output enable	25
		Input	24
C5	P45	Output	23
		Output enable	22
		Input	21
D5	P44	Output	20
		Output enable	19
		Input	18
C4	P43	Output	17
		Output enable	16
		Input	15
A4	P42	Output	14
		Output enable	13
		Input	12
B4	P41	Output	11
		Output enable	10
		Input	9
B3	P40	Output	8
		Output enable	7
		Input	6

Table 47.7 Boundary Scan Register (145-Pin TFLGA) (8/8)

Pin No.	Pin Name	Input/Output	Bit Name
B2	P07	Output	5
		Output enable	4
		Input	3
		To TDO	

47.3 Operations

The boundary scan functionality is valid when the RES# pin is driven high, EMLE pin is driven low, and the BSCANP pin is driven high.

47.3.1 TAP Controller

Figure 47.2 shows the state transition diagram of the TAP controller.

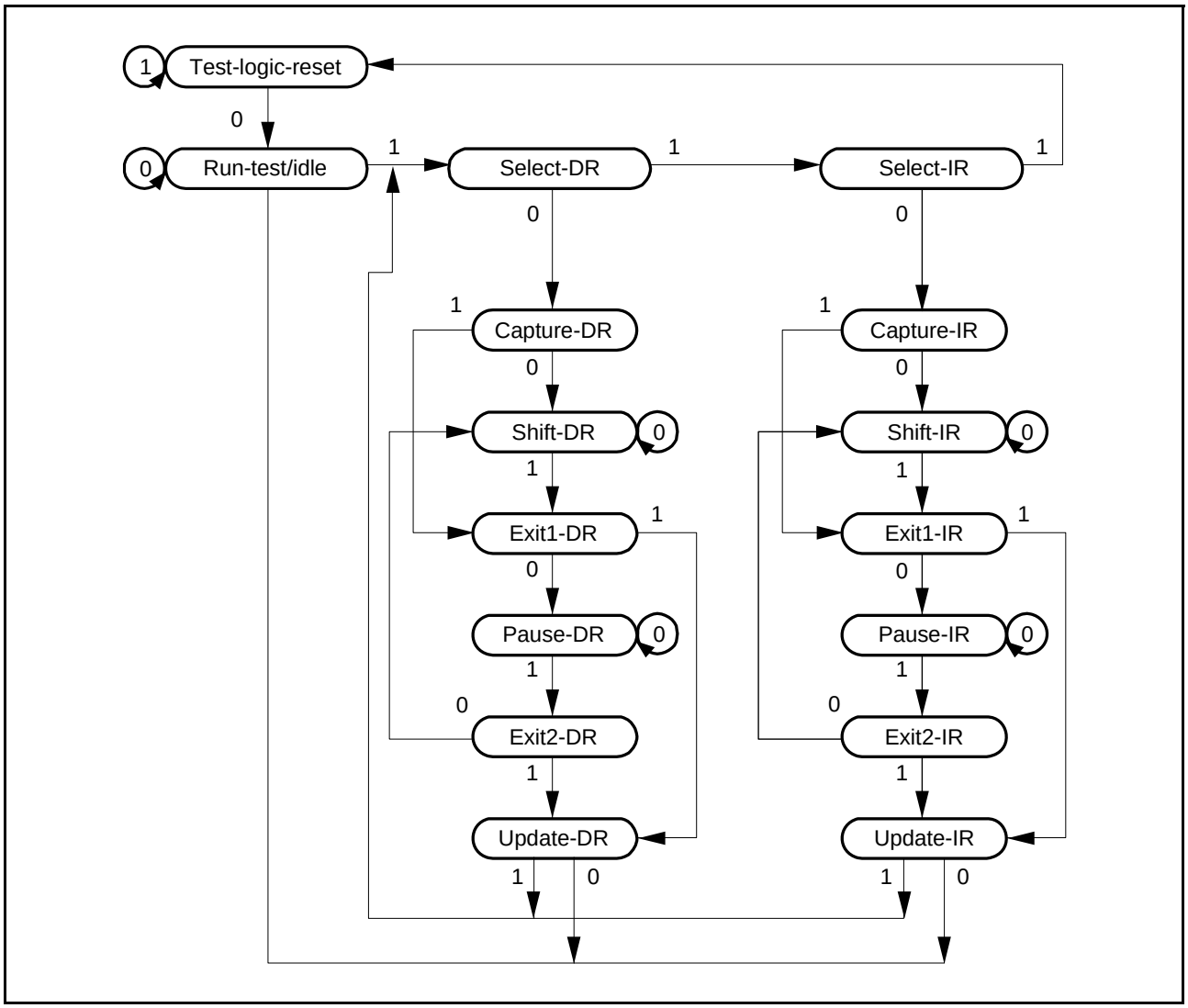


Figure 47.2 State Transition of TAP Controller

47.3.2 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. INSTR 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.

47.4 Usage Notes

(1) PIn serial transfer, data are input or output in LSB order (see Figure 47.3).

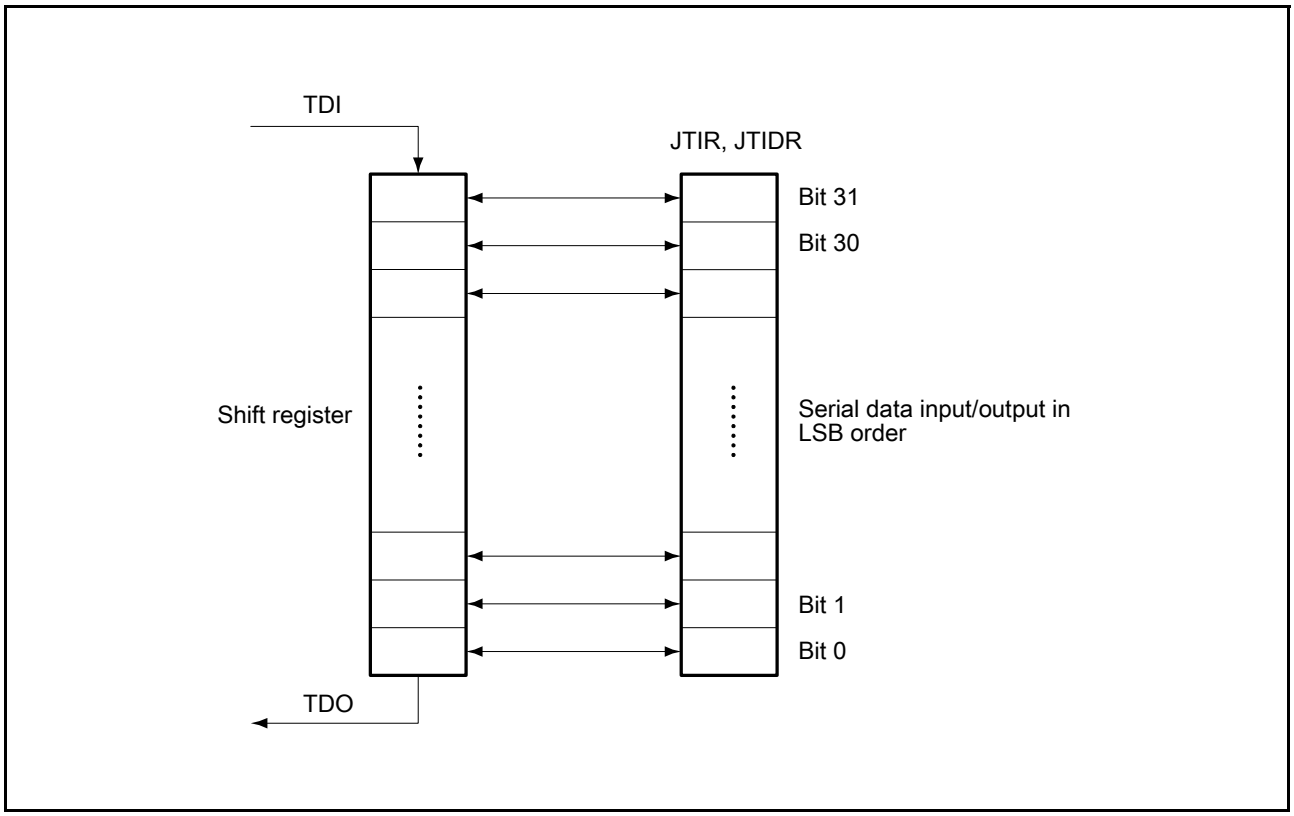


Figure 47.3 Serial Data Input/Output

(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, AVSS0, VREFH0, VREFL0, VREFH, VREFL, VCC_USB, and VSS_USB) cannot be boundary-scanned.

(4) Clock pins (EXTAL, XTAL, XCIN, and XCOUT) cannot be boundary-scanned.

(5) Reset signal (RES#) cannot be boundary-scanned.

(6) USB dedicated pins (USB0_DP and USB0_DM) cannot be boundary-scanned.

(7) The on-chip emulator enable pin (EMLE) cannot be boundary-scanned.

(8) The boundary-scan pin (BSCANP) cannot be boundary-scanned.

(9) The boundary-scan pins (TCK, TMS, TRST#, TDI, and TDO) cannot be boundary-scanned.

(10) The boundary-scan facility is not available when the chip is in the states below.

- Reset state
- Software standby or deep software standby

(11) 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.

(12) Figure 47.4 (1)shows the pin configuration of pins P12, P13, P20, and P21. When the boundary scan function is used with pins P12, P13, P16, P17, P20, P21, PC0, and PC1 to be used as RIIC pins (SCL0[FM+], SCL1 to SCL3, SDA0[FM+], and SDA1 to SDA3), the conflict with open-drain output or sneak current might be generated.

(13) Figure 47.4 (2) shows the pin configuration of pins P00 to P02, P40 to P47, P90 to P93, PD0 to PD7, and PE2 to PE7. When the boundary scan function is used with pins P00 to P02, P40 to P47, P90 to P93, PD0 to PD7, and PE2 to PE7 to be used as AD input pins (AN000 to AN020, and AN0 to AN7), the conflict with the AD input or sneak current might be generated.

(14) Figure 47.4 (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.

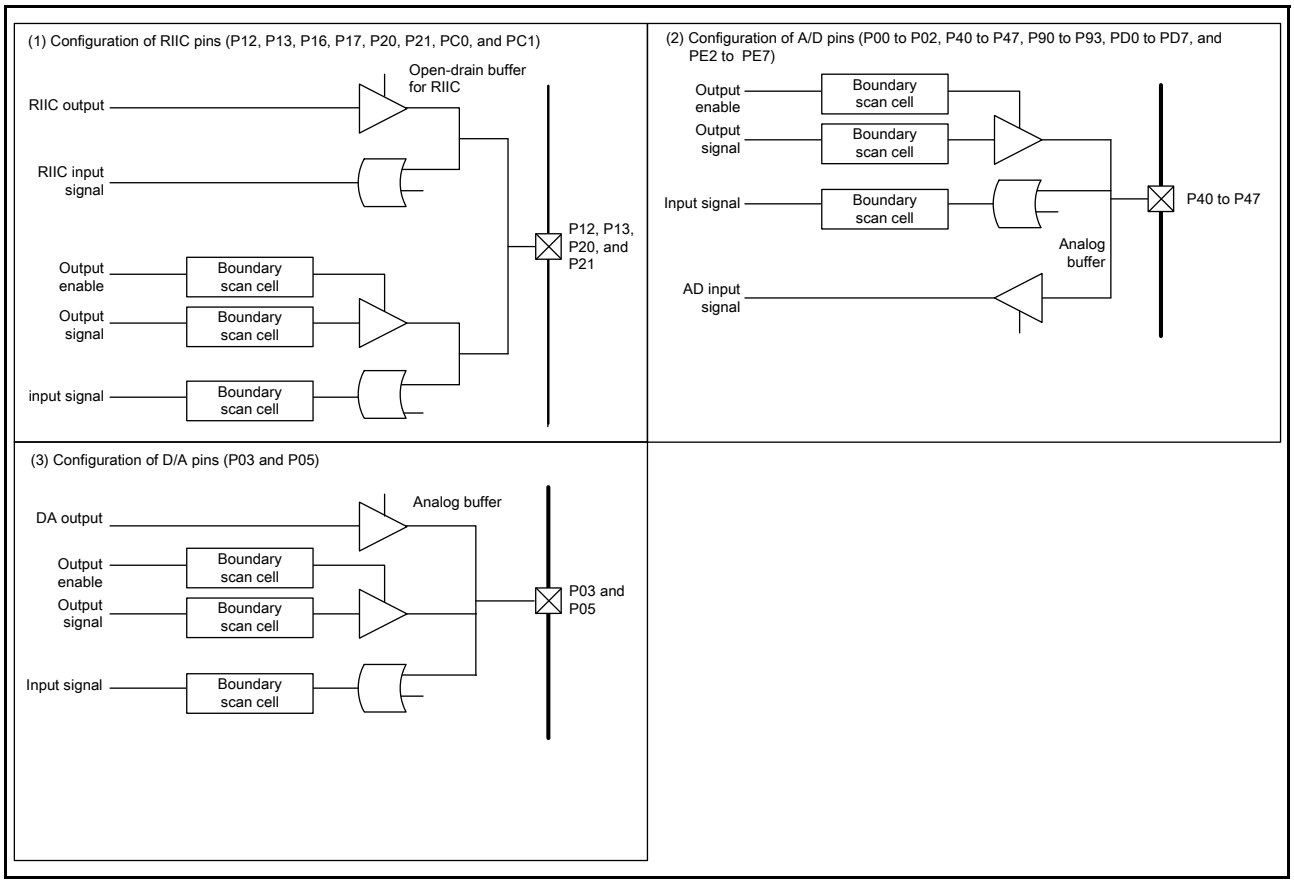


Figure 47.4 Pin Configuration

48. ROMless

This section focuses on differences between the specifications for ROMless versions and versions with flash memory. Other specifications are the same as those of the flash memory version, so refer to the corresponding sections for details.

48.1 Specification Differences between ROMless Versions and Versions with Flash Memory

Specifications for the following items differ between ROMless versions and versions with flash memory.

- Operating mode
- Address spaces
- Option-setting memory
- Clock generation circuit
- External bus
- Multi-function pin controller (MPC)
- ROM (flash memory for code storage)
- E2 DataFlash (flash memory for storing code)

Table 48.1 Specification Differences from Flash Memory Versions

Module/Function	ROMless Version
Operating mode	Supports only on-chip ROM disabled expansion mode. Supports only little endian.
Address space	Supports only on-chip ROM disabled expansion mode. Supports external address space.
Option setting memory	Option setting memory not installed. Supports functions corresponding to option setting memory except some functions.
Clock generation circuit	Does not generate FCLK which is supplied to the FlashIF.
Interrupt controller (ICUb)	No interrupts from the FCU.
External bus	The bus width of the CS0 area is fixed to 16 bits. Connectable pins are also fixed.
Multi-function pin controller (MPC)	Changing or rewriting the PFCSE register initial value is prohibited.
RAM	Lower 128 bytes are unavailable.
ROM (flash memory for storing code)	Not installed
E2 data flash (flash memory for storing data)	Not installed

48.2 Operating Mode

The ROMless version handles on-chip ROM disabled expansion mode but not single-chip mode and on-chip ROM enabled expansion mode.
Transitions to other modes are not possible after a transition to on-chip disabled expansion mode.
The MD pin must be set low when the emulator is used and high when the emulator is not used.

48.2.1 Endian

The endian is fixed as little endian.

48.2.2 Mode Status Register (MDSR)

Address(es): 0008 0002h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	—	Reserved	These bits are always read as 0 and cannot be modified.	R
b1	—	Reserved	These bits are always read as 1 and cannot be modified.	R
b15 to b2	—	Reserved	These bits are always read as 0 and cannot be modified.	R

Although MDSR is a readable and writable register, data written are invalid.
The EXB bit and IROM bit are read as 1.

48.2.3 System Control Register 0 (SYSCR0)

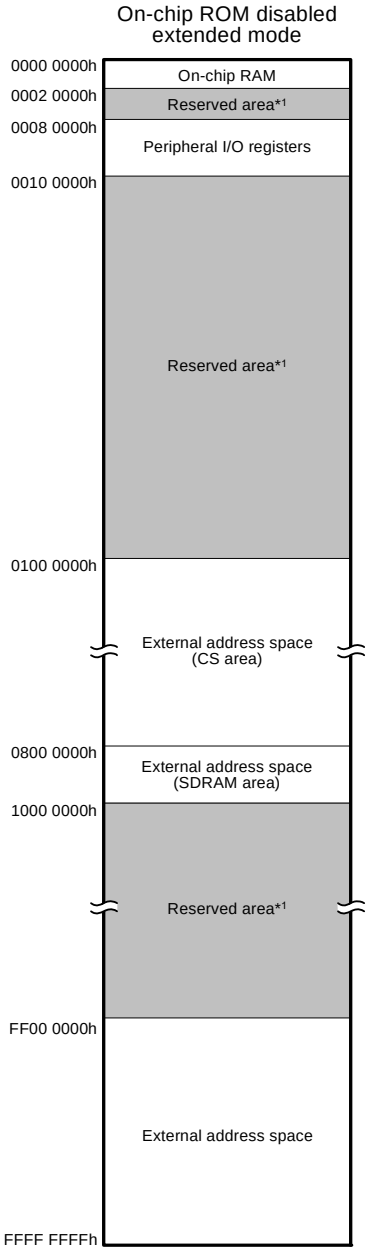
Address(es): 0008 0006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EXBE	ROME
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	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
b15 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R

48.3 Memory Map

The only available address mapping for the ROMless version is that for on-chip disabled expansion mode.



Note 1. Reserved area should not be accessed.

Figure 48.1 Memory Map in On-chip ROM Disabled Extended Mode

48.4 Option-Setting Memory

The ROMless version does not have option-setting memory and the user cannot rewrite it.
The values of the OFS0, OFS1n and MDES registers are fixed to FFFF FFFFh, FFFF FFFBh, and FFFF FFFh, respectively. See section 7, Option-Setting Memory, for the description of these registers.

48.5 Clock Generation Circuits

The flash interface clock (FCLK), which is for supply to the flash interface, is not generated.

48.5.1 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
	—	—	—	—	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]				—	—	—	—	—	—	—	—
Value after 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	The write value should always be 0000 0000b.	R/W
b11 to b8	PCKB[3:0]	Peripheral module clock B (PCLKB) select bits*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 Setting other than above is prohibited.	R/W
b15 to b12	PCKA[3:0]	Peripheral module clock A (PCLKA) select bits*2	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 Setting other than above is prohibited.	R/W
b19 to b16	BCK[3:0]	External bus clock (BCLK) select bits *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 Setting other than above is prohibited.	R/W
b21 to b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22	PSTOP0	SDCLK pin output control bit	0: Enables the SDCLK pin output. 1: Disables the SDCLK pin output (fixed to high level).	R/W
b23	PSTOP1	BCLK pin output control bit*3	0: Enables the BCLK pin output. 1: Disables the BCLK pin output (fixed to high level).	R/W

Bit	Symbol	Bit Name	Description	R/W
b27 to b24	ICK[3:0]	System clock (ICLK) select bits*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 Setting other than above is prohibited.	R/W
b31 to b28	—	Reserved	The write value should be 0010b.	R/W

Note 1. Setting 1/1 is prohibited when PLL is selected.
Note 2. Do not set frequency lower for ICLK than that of the external bus clock.
Note 3. P53 which is multiplexed with BCLK cannot be used as an I/O port when the external bus is enabled.
Note 4. Only a frequency division ratio of 1/1 is allowed for ICLK and FCLK when low-speed operating mode 2 is selected and the sub-clock oscillator is selected by the SCKCR3.CKSEL[2:0] bits.

48.6 FCU-related Interrupts

The FCU-related interrupt request enable register 02 (IER02) and interrupt source priority register n (IPRn) (n = 001 and 002) do not have flash interface error interrupt (FIFERR) and flash ready interrupt (FRDYI). Therefore, there are differences in the functions of the registers.

48.6.1 Interrupt Request Enable Register 02 (IER02)

Address(es): 0008 7202h

	b7	b6	b8	b4	b3	b2	b1	b0
	—	IEN6	—	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 bit 0	0: Disables interrupt requests 1: Enables interrupt requests.	R/W
b1	IEN1	Interrupt request enable bit 1		R/W
b2	IEN2	Interrupt request enable bit 2		R/W
b3	IEN3	Interrupt request enable bit 3		R/W
b4	IEN4	Interrupt request enable bit 4		R/W
b5	—	Reserved	This bit is read as 0. The write value should always be 0.	R/W
b6	IEN6	Interrupt request enable bit 6	0: Disables interrupt requests 1: Enables interrupt requests.	R/W
b7	—	Reserved	This bit is read as 0. The write value should always be 0.	R/W

48.6.2 Interrupt Source Priority Register n (IPRn) (n = 001, 002)

Address(es): 0008 7301h, 0008 7302h

	b7	b6	b8	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	This bit is read as 0. The write value should always be 0.	R/W

48.7 External Bus

48.7.1 CS0 Area

Processing by the ROMless version jumps to the first address in the CS0 area after release from the reset state, and booting up from the external boot ROM thus starts. Accordingly, mount the boot ROM in the CS0 area.
The bus width for use with the CS0 area is fixed to 16 bits. For a 8-bit bus or 32-bit bus, use another area.
On the ROMless version, the pin functions available in the CS0 area are fixed. When connecting external I/O or memory to the CS0 area, use the pins with the allocation listed in Table 48.2. Pin functions that are not listed in Table 48.2 are still available (e.g., WAIT#, WR1#, etc.). For example, the WAIT# pin function is selectable on port pins P51, P55, P57, and PC5. The fixed allocation of pin functions listed in Table 48.2 is invalid when using CS1 to CS7 areas. In this case, the required pin functions must be selected by pin assignment.

Table 48.2 List of Pin Functions with a Fixed Allocation When the CS0 Area Is in Use

Pin No.	Fixed Allocation	Pin Function	Pin No.	Fixed Allocation	Pin Function
76	PC7	A23	112	PA2	A2
77	PC6	A22	114	PA1	A1
78	PC5	A21	125	PE7	D15
82	PC4	A20	126	PE6	D14
83	PC3	A19	130	PE5	D13
86	PC2	A18	131	PE4	D12
89	PC1	A17	132	PE3	D11
91	PC0	A16	133	PE2	D10
94	PB7	A15	134	PE1	D9
95	PB6	A14	135	PE0	D8
96	PB5	A13	143	PD7	D7
97	PB4	A12	145	PD6	D6
98	PB3	A11	147	PD5	D5
99	PB2	A10	148	PD4	D4
100	PB1	A9	150	PD3	D3
104	PB0	A8	154	PD2	D2
106	PA7	A7	156	PD1	D1
107	PA6	A6	158	PD0	D0
108	PA5	A5	141	P60	CS0#*
109	PA4	A4	70	P52	RD#*
110	PA3	A3	72	P50	WR0#*

Note: • CS0#, RD#, WR0# is in the high-impedance state after reset cancelation. Therefore, external pull-up resistor should be added.

48.7.2 CS0 Control Register (CS0CR)

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	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EXENB	Operation enable	0: Disable 1: Enable	R/W*1
b3 to b1	—	Reserved*1	These bits are read as 0. The write value should always be 0.	R/W
b5 to b4	BSIZE[1:0]	External bus width select bits	b11 b8 0 0: Set to the 16-bit bus space. 0 1: Set to the 32-bit bus space. 1 0: Set to the 8-bit bus space. 1 1: Setting prohibited.	R/W*2
b7 to b6	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W
b8	EMODE	Endian mode	0: The endian for area n is identical to that for the operating mode. 1: The endian or area n differs from that for the operating mode.(n = 0 to 7)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W
b12	MPXEN	Address/data multiplexed I/O interface select	0: Area n is for the separate bus interface. 1: Area n is for the address/data multiplexed I/O interface.(n = 0 to 7)	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W

Note 1. The initial value of the CS0CR.EXENB bit is 1 and that of the CSnCR.EXENB (n = 1 to 7) bits is 0.
Note 2. The initial value of the CS0CR.BSIZE[1:0] bits is 10b.

48.8 Multi-Function Pin Controller (MPC)

48.8.1 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	1

Bit	Symbol	Bit Name	Description	R/W
b0	CS0E	CS0 enable	0: sables 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

TWhen writing to PFCSE, write 1 to the CS0E bit.

48.8.2 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	1	1	1	1	1	1	1	1

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

The PFAOE0 register must not be rewritten from the value after reset.

48.8.3 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	1	1	1	1	1	1	1	1

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

The PFAOE1 register must not be rewritten from the value after reset.

48.8.4 External Bus Control Register 0 (PFBCR0)

Address(es): 0008 C106h

	b7	b6	b5	b4	b3	b2	b1	b0
	WR32BC32E	WR1BC1E	DH32E	DHE	—	—	ADRHMS	ADRLE
Value after reset	0	0	0	1	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	ADRLE	A0 to A7 output enable	0: Sets PA0 to PA7 as the I/O ports. 1: Sets PA0 to PA7 as the external address bus A0 to A7.	R/W
b1	ADRHMS	A16 to A23 output enable	0: Sets PC0 to PC7 as the external address bus A16 to A23. 1: Sets P90 to P97 as the external address bus A16 to A23.	R/W
b3 to b2	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W
b4	DHE	D8 to D15 output enable	0: Sets PE0 to PE7 as the I/O ports. 1: Sets PE0 to PE7 as the external data bus D8 to D15.	R/W
b5	DH32E	D16 to D31 output enable	0: Sets PG0 to PG7 and P90 to P97 as the I/O ports. 1: Sets PG0 to PG7 and P90 to P97 as external data bus D16 to D31.	R/W
b6	WR1BC1E	WR1#/BC1# output enable	0: Sets P51 as the I/O port. 1: Sets P51 as either WR1# or BC1#.	R/W
b7	WR32BC32E	WR3#/BC3# output enable WR2#/BC2# output enable	0: Set P56 and P57 as the I/O ports. 1: Sets P56 as WR#2 or BC2#, and P57 as WR3# or BC3#.	R/W

The PFBCR0 register must not be rewritten from the value after reset.

48.9 RAM

The 128 bytes of RAM from 0001 FF7Fh to 0001 FFFFh are not available for use.

48.10 ROM

Flash memory for code storage is not installed.

48.11 E2 DataFlash

E2 DataFlash is not installed.

48.11.1 Usage Note

Bus pins are in the high-impedance state over a certain period following a reset. Therefore, add pull-up resistors for the CS0#, RD#, and WR0# control signals to avoid malfunctions of external devices on the external bus.

49. Electrical Characteristics

Appendix 1.Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1/3)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP=1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE = 1	OPE = 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	Hi-Z	[DAOE0=1] DA output retained [DAOE0=0] Keep-O*2		[DAOE0=1] Hi-Z [DAOE0=0] Keep	Keep	Hi-Z
P05/DA1/IRQ13	All	Hi-Z	[DAOE1=1] DA output retained [DAOE1=0] Keep-O*2		[DAOE1=1] Hi-Z [DAOE1=0] Keep	Keep	Hi-Z
P07/IRQ15	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P10/IRQ0, P11/IRQ1, P12/IRQ2, P13/IRQ3, P14/IRQ4 P15/IRQ5/CRX1, P16/IRQ6/SCL2, P17/IRQ7/SDA2	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P20/IRQ8, P21/IRQ9 P22, P23	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
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)		[CS output] H [Other than the above] Keep-O	[CS output] Hi-Z [Other than the above] Keep-O			
P30/IRQ0, P31/IRQ1, P32/IRQ2, P33/IRQ3 P34/IRQ4, 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, P41/IRQ9, P42/IRQ10, P43/IRQ11, P44/IRQ12, P45/IRQ13, P46/IRQ14, P47/IRQ15	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	[WR0#/WR# output] Hi-Z			
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	[RD# output] Hi-Z			
P53/BCLK	All	Hi-Z	[Clock output] H [Other than the above] Hi-Z		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/WR2#/BC2#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[WR2#/BC2# output] H [Other than the above] Keep-O	[WR2#/BC2# output] Hi-Z [Other than the above] Keep-O			
P57/WR3#/BC3#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[WR3#/BC3# output] H [Other than the above] Keep-O	[WR3#/BC3# output] Hi-Z [Other than the above] Keep-O			

Table 1.1 Port States in Each Processing State (2/3)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP=1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE = 1	OPE = 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)		[CS output] H [Other than the above] Keep-O	[CS 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)		[CS output] H [SDCS# output] SDCS# output retained [Other than the above] Keep-O	[CS output] Hi-Z [SDCS# output] Hi-Z [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)		[CS output] H [RAS# output] RAS# output retained [Other than the above] Keep-O	[CS output] Hi-Z [RAS# output] Hi-Z [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)		[CS output] H [CAS# output] CAS# output retained [Other than the above] Keep-O	[CS output] Hi-Z [CAS# output] Hi-Z [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)		[CS output] H [WE# output] WE# output retained [Other than the above] Keep-O	[CS output] Hi-Z [WE# output] Hi-Z [Other than the above] Keep-O			
P65/CS5#/CKE	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS output] H [CKE output] CKE output retained [Other than the above] Keep-O	[CS output] Hi-Z [CKE output] Hi-Z [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)		[CS output] H [DQM0 output] DQM0 output retained [Other than the above] Keep-O	[CS output] Hi-Z [DQM0 output] Hi-Z [Other than the above] Keep-O			
P67/CS7#/DQM1/IRQ15	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS output] H [DQM1 output] DQM1 output retained [Other than the above] Keep-O	[CS output] Hi-Z [DQM1 output] Hi-Z [Other than the above] Keep-O			
P70/SDCLK	All	Hi-Z	[Clock output] H [Other than the above] Hi-Z		Keep	Keep	Hi-Z
P71/CS1#, P72/CS2#, P73/CS3#, P74/CS4#, 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)		[CS output] H [Other than the above] Keep-O	[CS output] Hi-Z [Other than the above] Keep-O			
P80/TRDATA0, P81/TRDATA1, P82 to 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			

Table 1.1 Port States in Each Processing State (3/3)

Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP=1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
				OPE = 1	OPE = 0		IOKEEP = 1*1	IOKEEP = 0
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*2	[Address output] Hi-Z [Other than the above] Keep-O*2			
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			
PA2, PA3/IRQ6, 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			
PB0/IRQ12, PB1/IRQ4, 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/CS1#/IRQ13, PC7/CS0#/IRQ14	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			
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 [Other than the above] Keep-O*2	[Data output] Hi-Z [Other than the above] Keep-O*2			
PE0, PE1, PE2/IRQ7, 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 [Other than the above] Keep-O*2	[Data output] Hi-Z [Other than the above] Keep-O*2			
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		Keep-O*3	Hi-Z	
USB0_DP	All		Hi-Z	Keep-O*4		Keep-O*3	Hi-Z	
USB1_DM	All		Hi-Z	Keep-O*4		Keep-O*3	Hi-Z	
USB1_DP	All		Hi-Z	Keep-O*4		Keep-O*3	Hi-Z	

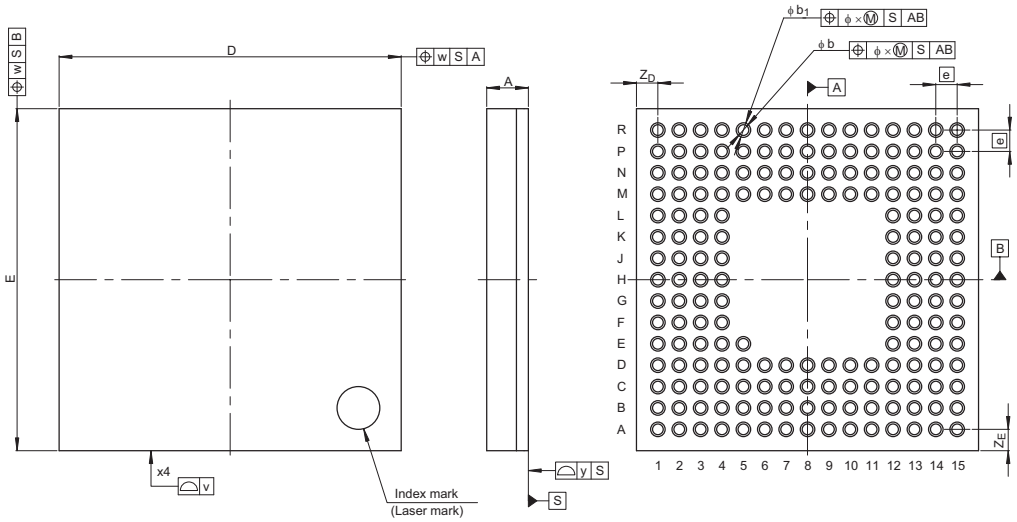
H: High-level
L: Low-level
Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Keep: Pin states are retained during periods on software standby.
Hi-Z: High-impedance

Note 1. Ratins 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.

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.

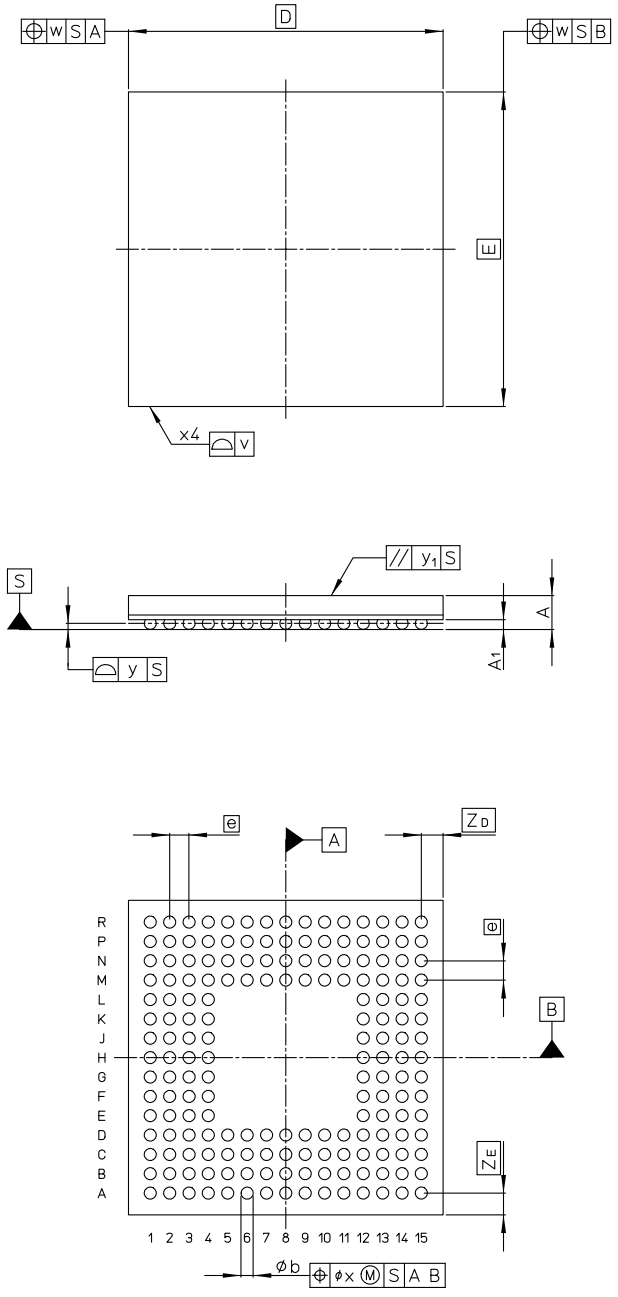
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA177-8x8-0.50	PTLG0177KA-A	177F0E-A	0.2g



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)

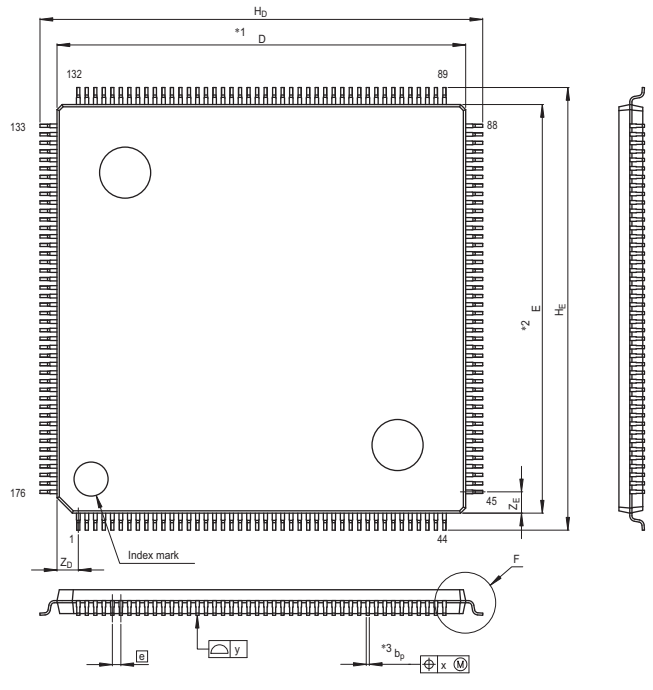
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA176-13x13-0.80	PLBG0176GA-A	BP-176/BP-176V	0.45g

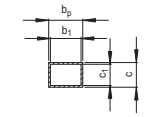


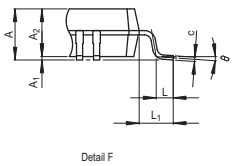
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	13.0	—
E	—	13.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y1	—	—	0.2
SD	—	—	—
SE	—	—	—
ZD	—	0.90	—
ZE	—	0.90	—

Figure B 176-pin LFBGA (PLBG0176GA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP176-24x24-0.50	PLQP0176KB-A	176P6Q-A / FP-176E / FP-176EV	1.8g





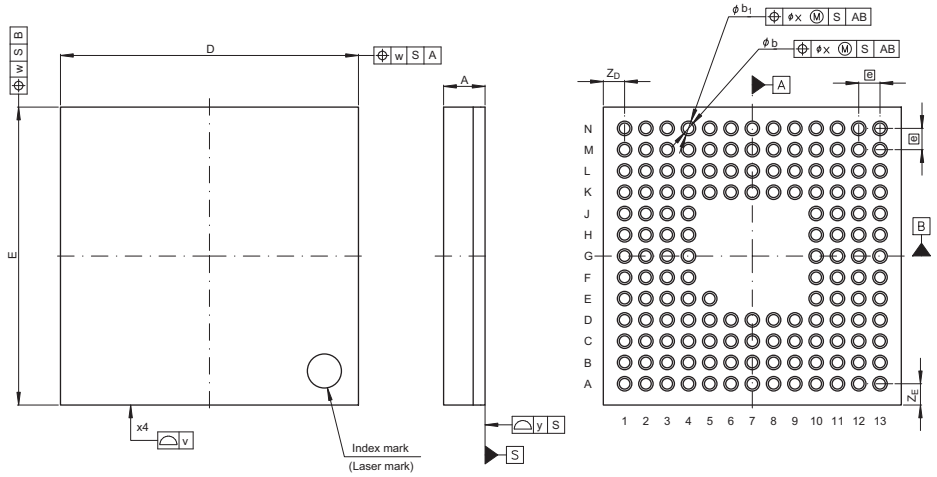


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

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 LQFP (PLQP0176KB-A)

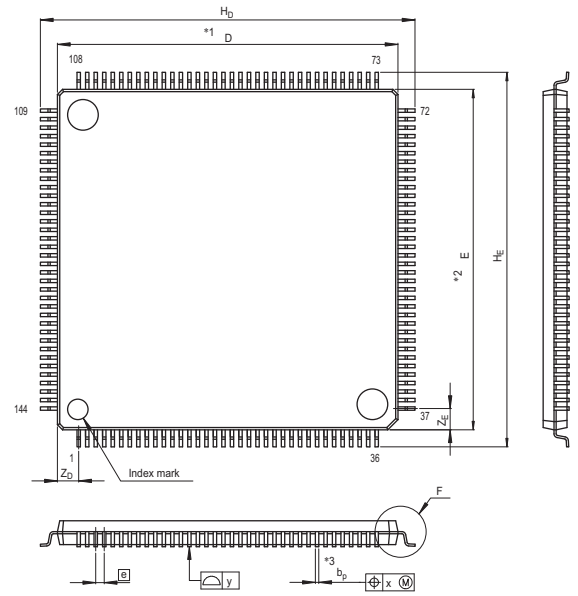
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA145-7x7-0.50	PTLG0145KA-A	145F0G	0.1g

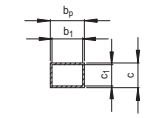


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	7.0	—
E	—	7.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.5	—
b	0.21	0.25	0.29
b_1	0.29	0.34	0.39
x	—	—	0.08
y	—	—	0.10
Z_D	—	0.5	—
Z_E	—	0.5	—

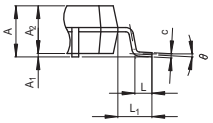
Figure D 145-pin TFLGA (PTLG0145KA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g





Terminal cross section



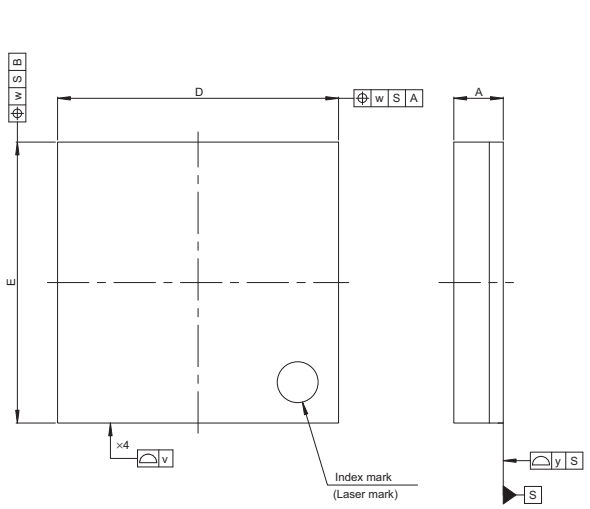
Detail F

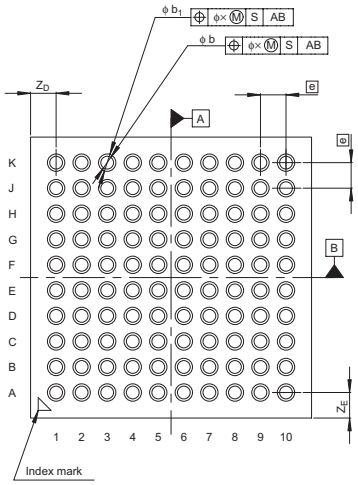
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

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 LQFP (PLQP0144KA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA100-5.5x5.5-0.50	PTLG0100KA-A	100F0M	0.1g

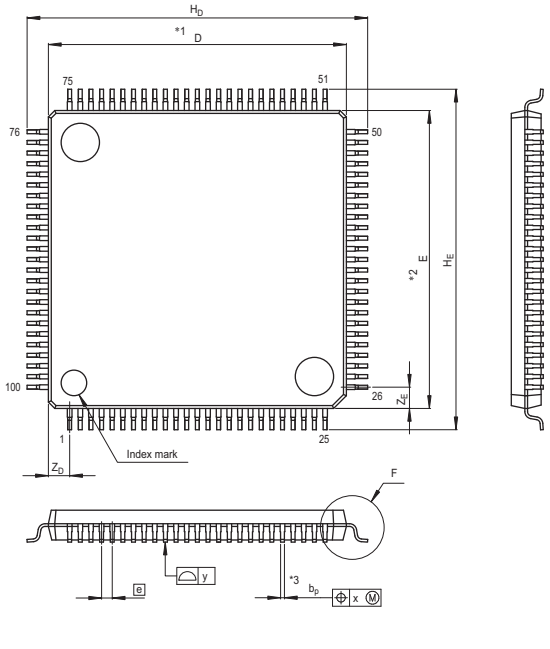


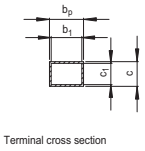


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	5.5	—
E	—	5.5	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.5	—
b	0.21	0.25	0.29
b_1	0.29	0.34	0.39
x	—	—	0.08
y	—	—	0.10
Z_D	—	0.5	—
Z_E	—	0.5	—

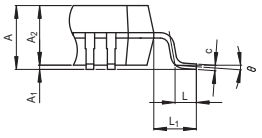
Figure F 100-pin TFLGA (PTLG0100KA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g





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	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 LQFP (PLQP0100KB-A)

REVISION HISTORY	RX63N Group, RX631 Group User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
0.50	May 12, 2011	—	First Edition issued
0.90	Dec 12. 2011	All	
		—	Package added (177-pin TFLGA, 176-pin LFBGA, 145-pin TFLGA), module name changed
		—	Interrupt Controller (ICUb) module name changed
		1. Overview	
		50 to 54	Table 1.1 Outline of Specifications, Reset, Realtime clock, Temperature sensor, Power supply voltage, changed
		56 to 58	Table 1.3 List of Products, changed
		58	Figure 1.1 How to Read the Product Part No., changed
		60 to 65	Table 1.4 Pin Functions, BSCANP pin added
		66	Figure 1.3 Pin Assignment (176-Pin TFLGA), added
		67	Figure 1.4 Pin Assignment (176-Pin LFBGA), added
		68	Figure 1.5 Pin Assignment (176-Pin LQFP), pin 18 changed
		69	Figure 1.6 Pin Assignment (144-Pin TFLGA), added
		70	Figure 1.7 Pin Assignment (144-Pin LQFP), pin 16 changed
		71	Figure 1.8 Pin Assignment (100-Pin LQFP), pin 7 changed
		72 to 76	Table 1.5 List of Pins and Pin Functions (177-pin TFLGA, 176-pin LFBGA), added
		82 to 86	Table 1.7 List of Pins and Pin Functions (145-Pin TFLGA), added
		2. CPU	
		116	2.8.2 Instructions and Pipeline Processing, changed
		116	Table 2.13 Instructions that are Converted into a Single Micro-Operation, changed
		118 to 119	Table 2.14 Instructions that are Converted into Multiple Micro-Operations, changed
		3. Operating Modes	
		124	Table 3.1 Selection of Operating Modes by the Mode Pin, Notes 1 and 2, changed
		124	Table 3.2 Selection of Operating Modes by Register Setting, Note 1, changed
		124	Table 3.3 Endian Setting Method, changed
		124	Table 3.4 Selection of Endian, changed
		126	3.2.3 System Control Register 0 (SYSCR0), Bit table (KEY[7:0]), Description, changed
		128	3.3.5 User Boot Mode, changed
		5. I/O Registers	
		134	(1) I/O register addresses (address order), changed
		135	(3) Number of Access Cycles to I/O Registers, changed
		136 to 181	Table 5.1 List of I/O Registers, changed
		6. Resets	
		All	pin reset → reset from a pin
		183	Table 6.2 Targets to be Initialized by Each Reset Source, changed
		188	6.3.2 Power-On Reset and Voltage Monitoring 0 Reset, changed
		189	6.3.3 Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset, changed
		190	Figure 6.2 Operation Examples During Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset, changed
		191	6.3.4 Deep Software Standby Reset, changed
		191	6.3.5 Independent Watchdog Timer Reset, changed
		191	6.3.6 Watchdog Timer Reset, changed
		191	6.3.7 Software Reset, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	7. Option-Setting Memory	
		199	7.2.2 Option Function Select Register 1 (OFS1), Note 1 for the bit table deleted, register description changed
		201	7.3 UB Code, changed
		8. Voltage Detection Circuit (LVDA)	
		202	8.1 Overview, changed
		202	Table 8.1 Specifications of Voltage Detection Circuit, changed
		204	Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit, changed
		204	Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit, changed
		206	8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR), bit table (LVD1MON) description changed, LVD1DET flag description changed
		208	8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR), bit table (LVD2MON), description changed, LVD2DET flag description changed
		210	8.2.6 Voltage Detection Level Select Register (LVDLVLR), the figure of bit assignment, value after reset changed, Description of bit table changed, Register description changed
		211	8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0), description of LVD1RIE, LVD1DFDIS, and LVD1FSAMP[1:0], changed
		213	8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0), description of LVD2RIE, LVD2DFDIS, and LVD2FSAMP[1:0] changed
		215	8.3.2 Monitoring Vdet1, changed
		215	8.3.3 Monitoring Vdet2, changed
		217	8.5 Interrupt and Reset from Voltage Monitor 1, changed
		217	Table 8.2 Procedures for Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset so that Voltage Monitoring Operates, changed
		217	Table 8.3 Procedures for Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset to Stop Voltage Monitoring, added
		218	Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation, changed
		219	8.6 Interrupt and Reset from Voltage Monitor 2, changed
		219	Table 8.4 Procedures for Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset so that Voltage Monitoring Operates, changed
		219	Table 8.5 Procedures for Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset to Stop Voltage Monitoring, added
		220	Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation, changed
		9. Clock Generation Circuit	
		All	High-speed clock oscillator → High-speed on-chip oscillator, Low-speed clock oscillator → Low-speed on-chip oscillator IWDT-dedicated low-speed oscillator → IWDT-dedicated on-chip oscillator High speed clock → HOCO clock, Low speed lock → LOCO clock
		221 to 222	Table 9.1 Specifications of Clock Generation Circuit, operating frequency and sub-clock oscillator, changed, Note 1 added
		224	9.2.1 System Clock Control Register (SCKCR), bit table (b3 to b0, b7 to b4, b15 to b12) description changed, Note1 to Note 4 changed
		225	9.2.2 System Clock Control Register 2 (SCKCR2), bit table (UCK[3:0]) description, changed
		228	9.2.4 PLL Control Register (PLLCR), bit table (PLIDIV[1:0]) description changed, register description changed
		229	9.2.5 PLL Control Register 2 (PLLCR2), description of PLLEN, changed
		231	9.2.7 Main Clock Oscillator Control Register (MOSCCR), description of MOSTP, changed
		232	9.2.8 Sub-Clock Oscillator Control Register (SOSCCR), description of SOSTP, changed
		233	9.2.9 Low-Speed Clock Oscillator Control Register (LOCOCR), description of LCSTP, changed
		234	9.2.10 IWDT-Dedicated On-Chip Clock Oscillator Control Register (ILOCOCR), description of ILCSTP, changed
		235	9.2.11 High-Speed On-Chip Oscillator Control Register (HOCOCR), description of HCSTP, changed
		236	9.2.12 Oscillation Stop Detection Control Register (OSTDCR), description of OSTDIE, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	238	9.2.14 Main Clock Oscillator Forced Oscillation Control Register (MOFCR), description of MOFXIN, changed
		239	9.2.15 High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR), description of HOCOPCNT, changed
		240	Figure 9.2 Example of Crystal Resonator Connection, changed
		240	Table 9.3 Damping Resistance (Reference Values), changed
		240	Figure 9.3 Equivalent Circuit of Crystal Resonator, changed
		240	Table 9.4 Crystal Resonator Characteristics (Reference Values), changed
		241	9.3.2 External Clock Input, changed
		241	9.3.3 Notes on the External Clock Input, added
		242	Figure 9.5 Connection Example of 32.768-kHz Crystal Resonator, changed
		242	Table 9.5 Crystal Resonator Characteristics (Reference Values), changed
		242	9.4.2 Handling of Pins when Sub-Clock is Not Used, changed
		244	Figure 9.8 Flow of Recovery from Detection of Oscillator Stop, added
		244	9.5.2 Oscillation Stop Detection Interrupts, changed
		245	9.7 Internal Clock, changed
		—	9.9 Pin Settings When an External Clock is Connected (2) Subclock, deleted
		247	9.10.1 Notes on Clock Generation Circuit (1), (2) changed
		248	9.10.4 Notes on Resonator Connect Pin, changed
		248	9.10.5 Notes on Sub-Clock, changed
		10. Frequency Measurement Circuit (MCK)	
		249	Table 10.1 Specifications of Frequency Measurement Circuit, changed
		250	Figure 10.1 Block Diagram of Frequency Measurement Circuit with MTU (System 1), changed
		250	Figure 10.2 Block Diagram of Frequency Measurement Circuit with TPU Unit 0 (System 2), changed
		254	Figure 10.3 Software Configuration, changed
		252	10.2.1 Counter-Clock Extension Register n, description of bit table (SCK[1:0]) changed, register description changed
		—	10.2.2 Counter-Clock Extension Register 2 (SCK2), deleted
		253	Figure 10.4 Operation Flow of Frequency Measurement Circuit, changed
		254	Figure 10.5 Example of MTU/TPU Operation, changed
		11. Low Power Consumption	
		256	11.1 Overview, changed
		256	Table 11.1 Specifications of Low Power Consumption Functions, changed
		265 to 266	Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode, Note 9 and Note 13, changed
		258	Figure 11.1 Mode Transitions, changed
		259	11.2.1 Standby Control Register (SBYCR), description of SSBY, changed
		265	11.2.5 Operating Power Control Register (OPCCR), description of OPCM[2:0], changed
		268	11.2.7 Main Clock Oscillator Wait Control Register (MOSCWTCR), register description, changed
		269	11.2.8 Sub-Clock Oscillator Wait Control Register (SOSCWTCR), description of bit table (SSTS[4:0]) changed, register description changed
		270	11.2.9 PLL Wait Control Register (PLLWTCR), register description, changed
		271	11.2.10 Deep Standby Control Register (DPSBYCR), description of DEEPCUT[1:0] and DPSBY, changed
		275	11.2.13 Deep Standby Interrupt Enable Register 2 (DPSIER2), description of bit table (DLVD1IE, DLVD2IE) changed
		286	11.3 Reducing Power Consumption by Switching Clock Signals, changed
		287	11.6.1.1 Transition to Sleep Mode, changed
		287	11.6.2.1 Transition to All-Module Clock Stop Mode, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	289	11.6.2.2 Canceling All-Module Clock Stop Mode, changed
		289	11.6.3.1 Transition to Software Standby Mode, changed, Note 1 changed
		290	11.6.3.2 Canceling Software Standby Mode, changed
		293	11.6.4.2 Canceling Deep Software Standby Mode, changed
		295	Figure 11.4 Example of Flowchart to Use Deep Software Standby Mode, changed
		296	11.7.7 Rewrite the Register by DMAC and DTC in Sleep Mode, added
		296	11.7.8 Transition to Software Standby Mode for Products with 1.5-Mbyte ROM or More or Products with 176 Pins or More, changed
		297	11.7.10 Point for Caution when Using the Sub-Clock as the Source of the System Clock, added
		12. Battery Backup Function	
		—	12.2 Register Descriptions, deleted
		—	12.2.1 Battery Backup Voltage Monitor Function Select Register (VBATTMNSELR), deleted
		—	12.2.2 Battery Backup Voltage Monitor Register (VBATTMONR), deleted
		299	12.2.1 Battery Backup Function, changed
		299	Figure 12.2 Operation for Switching to Battery Backup Function, changed
		—	12.3.2 VBATT Voltage Monitor, deleted
		—	Figure 12.3 Operation Example of VBATT Voltage Monitor, deleted
		300	12.3 Usage Notes, changed
		13. Register Write Protection Function	
		301	Table 13.1 Association between PRCR Bits and Registers to be Protected, PRC1 and PRC bits, changed
		302	13.1.1 Protect Register (PRCR), description of bit table (PRKEY[7:0]), changed
		15. Interrupt Control Unit (ICUb)	
		311	Table 15.1 Specifications of Interrupt Control Unit, changed
		312	Figure 15.1 Block Diagram of Interrupt Control Unit, changed
		324	15.2.13 Non-Maskable Interrupt Status Register (NMISR), bit name of OSTST, LVD1ST and LVD2ST, changed
		326	15.2.14 Non-Maskable Interrupt Enable Register (NMIER), bit name of OSTEN, LVD1EN and LVD2EN changed
		335	15.2.21 Group m Interrupt Clear Register (GCRm), bit name in the bit table of GCR12 register, modified
		338 to 344	Table 15.3 Interrupt Vector Table, Name modified (CUP, ALM, GROUP0 to GROUP6, GROUP12)
		354	Figure 15.15 1 Operation Example in Which Multiple Level Detection Interrupt Requests of a Group are Generated, changed
		360	15.6 Non-maskable Interrupt Operation, changed
		362	15.8.1 Point for Caution Regarding WAIT Instructions While the Non-Maskable Interrupt Is in Use, added
		16. Buses	
		363	Figure 16.1 Bus Configuration, Note1, added
		365	Table 16.2 Addresses Assigned for Each Bus, changed
		370	Table 16.6 Pin Configuration of the External Bus, description of WR1#, WR2# and WR3#, changed
		374	16.3.1 CSn Control Register (CSnCR), register description, changed
		375	16.3.2 CSn Recovery Cycle Register (CSnREC), register description, changed
		377	16.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN), register description, changed
		381	16.3.5 CSn Wait Control Register 1 (CSnWCR1), register description changed, notes of CSPWWAIT[2:0] and CSWWAIT[4:0] changed
		384	16.3.6 CSn Wait Control Register 2 (CSnWCR2), Note 1 for the bit table deleted, register description changed, Note of AWAIT[1:0] added, Notes of WRON[2:0] and WDON[2:0] changed
		387	16.3.8 SDC Mode Register (SDCMOD), register description, changed
		391	16.3.13 SDRAM Initialization Sequence Control Register (SDICR), register description, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	392	16.3.14 SDRAM Initialization Register (SDIR), register description, changed
		394	16.3.16 SDRAM Timing Register (SDTR), register description, changed
		396	16.3.17 SDRAM Mode Register (SDMOD), register description, changed
		415	16.5.1 Separate Bus, changed
		424	Figure 16.30 Example of Normal-Read Operation, changed
		438	Figure 16.47 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Normal Access), changed
		439	Figure 16.49 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), added
		440	Figure 16.50 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 where Bus Access is Divided up; m = 0 to 3), added
		440	Figure 16.51 Example of Recovery Cycle Insertion (with Address/Data Multiplexed I/O Interface; m = 0, 1), changed
		441	Figure 16.52 Example of Recovery Cycle Insertion When a Bus Access is Split (with Address/Data Multiplexed I/O Interface; m = 0, 1), changed
		442	Table 16.10 Limitations at the Time of Normal and Page Access, changed
		478	Table 16.21 Types of Bus Errors, changed
		17. DMA Controller (DMACA)	
		482	17.2.3 DMA Transfer Count Register (DMCRA), (3) Block Transfer Mode, changed
		483	17.2.4 DMA Block Transfer Count Register (DMCRB), changed
		499	17.3.2 Extended Repeat Area Function, changed
		502	Table 17.7 DMAC Execution Cycles, Note 1, added
		18. EXDMA Controller (EXDMACa)	
		516	Table 18.1 Specifications of EXDMAC, DMA request source, changed
		517	Figure 18.1 Block Diagram of EXDMAC, changed
		521	18.2.4 EXDMA Block Transfer Count Register (EDMCRB), description of bit table (b9-b0) changed, register description changed
		522	18.2.5 EXDMA Transfer Mode Register (EDMTMD), description of bit table (DCTG[1:0]) changed
		535	18.2.14 EXDMA External Request Flag Register (EDMERF), description of EREQ flag (clearing conditions), changed
		536	18.2.15 EXDMA Peripheral Request Flag Register (EDMPRF), description of PREQ flag, changed
		539	18.3.1 Transfer Mode (4) Cluster Transfer Mode, changed
		545	18.3.2 Extended Repeat Area Function, changed
		547	18.3.3 Address Update Function using Offset EXDMACn.EDMOFR → EXDMAC0.EDMOFR
		552	18.4 Transfer Operation, Note, added
		559	18.5.1 Activation Sources, changed
		567	18.7 Interrupts, changed
		568	Figure 18.26 EXDMAC Procedures for Suspending or Resuming DMA Transfer by the EXDMAC Interrupt, Title of the figure changed
		571	Figure 18.27 Operation Example in Normal-Transfer (CS Area Read) Single Address Mode, changed
		572	Figure 18.28 Operation Example in Normal-Transfer (CS Area Write) Single Address Mode, changed
		529	Figure 18.29 Operation Example in Normal-Transfer (SDRAM Area Read) Single Address Mode, changed
		574	Figure 18.30 Operation Example in Normal-Transfer (SDRAM Area Write) Single Address Mode, changed
		575	Figure 18.31 Operation Example in Block-Transfer (CS Area Read) Single Address Mode, changed
		576	Figure 18.32 Operation Example in Block-Transfer (CS Area Write) Single Address Mode, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	576	18.9.4 EDACKn Operation Example in Block-Transfer (SDRAM Area) Single Address Mode, changed
		577	Figure 18.33 Operation Example in Block-Transfer (SDRAM Area Read: Consecutive Access Enabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode, changed
		578	Figure 18.34 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Enabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode, changed
		579	Figure 18.35 Operation Example in Block-Transfer (SDRAM Area Read: Continuous Access Enabled, EXDMACn.EDMOMD.DACKSEL = 1) Single Address Mode, changed
		580	Figure 18.36 Operation Example in Block-Transfer (SDRAM Area Write: Continuous Access Enabled, EXDMACn.EDMOMD.DACKSEL = 1) Single Address Mode, changed
		581	Figure 18.37 Operation Example in Block-Transfer (SDRAM Area Read: Consecutive Access Disabled, EXDMACn.EDMOMD.DACKSEL:= 0) Single Address Mode, changed
		582	Figure 18.38 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Disabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode, changed
		19. Data Transfer Controller (DTCa)	
		587	19.2.2 DTC Mode Register B (MRB), description of bit table (CHNS), changed
		590	19.2.6 DTC Transfer Count Register B (CRB), register description, changed
		603	19.4.6 Chain Transfer, misdescription modified (MRB.CHNE bit → MRB.CHNS bit)
		604	Figure 19.9 Example (1) of DTC Operation Timing, changed
		604	Figure 19.10 Example (2) of DTC Operation Timing, changed
		605	Figure 19.11 Example (3) of DTC Operation Timing, changed
		605	Figure 19.12 Example (4) of DTC Operation Timing, changed
		606	Figure 19.13 Example of Operation when Transfer Information Skip is Executed, changed
		607	Table 19.8 Execution Cycles of the DTC, Note 7, added
		20. I/O Ports	
		All	NMOS open drain output → N channel open drain
		615	Table 20.1 Specifications of I/O Ports, changed
		616 to 617	Table 20.2 Port Functions, added
		618 to 621	20.2 I/O Port Configuration, added
		622	20.3.1 Port Direction Register (PDR), register description, changed
		623	20.3.2 Port Output Data Register (PODR), register description, changed
		625	20.3.4 Port Mode Register (PMR), register description, changed
		626	20.3.5 Open Drain Control Register 0 (ODR0), register description, changed
		627	20.3.6 Open Drain Control Register 1 (ODR1), register description, changed
		628	20.3.7 Pull-Up Resistor Control Register (PCR), register name, changed
		629	20.3.8 Drive Capacity Control Register (DSCR), register description, changed
		630	20.4 Handling of Unused Pins, added
		21. Multi-Function Pin Controller (MPC)	
		All	Register name changed, port mn pin function selection register → Pmn pin function control register
		632 to 647	Table 21.1 Functions Assigned to Each Multiplexed Pin, changed (Ethernet controller, SCI4, SCI9 and CAN2 changed, NMI added), Notes 1, 4 and 5 deleted and Note 3 changed
		648	21.2 Register Descriptions, changed
		648	21.2.1 Write-Protect Register (PWPR), description of PFSWE, changed
		678	21.2.25 External Bus Control Register 1 (PFBCR1), bit name and its description changed (MDSDE bit), description of DQM1E changed, Note 1 deleted, description of WAITS[1:0] and MDSDE added,
		682	21.3 How to Set the External Bus Interface, added
		685	21.4.2 Notes on MPC Register Setting (6), changed
		685 to 686	Table 21.28 Register Settings, changed
		686	21.4.3 Notes on the Use of Analog Functions, added

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	22. Multi-Function Timer Pulse Unit 2 (MTU2a)	
		688 to 689	Table 22.2 MTU Functions, changed
		690	Figure 22.1 Block Diagram of MTU, changed
		720	22.2.17 Timer Output Master Enable Registers (TOER), Note1, changed
		766	Figure 22.39 Counter Operation in Complementary PWM Mode, changed
		768	Figure 22.40 Example of Operation in Complementary PWM Mode, changed
		770	Figure 22.41 Example of Operation without Dead Time, changed
		771	Figure 22.42 Example of PWM Cycle Updating, changed
		772	Figure 22.43 Example of Data Updating in Complementary PWM Mode, changed
		773	Figure 22.44 Example of Initial Output in Complementary PWM Mode (1), changed
		774	Figure 22.45 Example of Initial Output in Complementary PWM Mode (2), changed
		775	Figure 22.46 Example of Waveform Output in Complementary PWM Mode (1), changed
		775	Figure 22.47 Example of Waveform Output in Complementary PWM Mode (2), changed
		776	Figure 20.48 Example of Waveform Output in Complementary PWM Mode (3), changed
		776	Figure 22.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1), changed
		777	Figure 22.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2), changed
		777	Figure 22.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3), changed
		778	Figure 22.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4), changed
		778	Figure 22.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5), changed
		779	Figure 22.55 Counter Clearing Synchronized with Another Channel, changed
		780	Figure 22.56 Timing for Synchronous Counter Clearing, changed
		781	Figure 22.58 Example of Synchronous Clearing in Dead Time during Up-Counting, changed
		782	Figure 22.59 Example of Synchronous Clearing in Interval Tb at Crest, changed
		782	Figure 22.60 Example of Synchronous Clearing in Dead Time during Down-Counting, changed
		783	Figure 22.61 Example of Synchronous Clearing in Interval Tb at Trough, changed
		786	Figure 22.68 Periods during which Interrupt Skipping Count can be Changed, changed
		788	Figure 22.70 Example of Operation when Buffer Transfer is Disabled (BTE1 = 0 and BTE0 = 1), changed
		788	Figure 22.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0), changed
		789	Figure 22.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period, changed
		799	22.4.3 A/D Converter Activation (4), changed
		808	22.6.2 Input Clock Restrictions, changed
		808	Figure 22.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode, changed
		817	Figure 22.121 Contention between Overflow and Counter Clearing, changed
		817	22.6.18 Contention between TCNT Write Operation and Overflow/Underflow, changed
		817	Figure 22.122 Contention between TCNT Write Operation and Overflow, changed
		818	22.6.23 Notes when Complementary PWM Mode Output Protection Functions are not Used, changed
		821	22.6.25 Continuous Output of Compare-Match Pulse Interrupt Signal, added
		23. Port Output Enable 2 (POE2a)	
		849	Figure 23.1 POE Block Diagram, changed
		850	Table 23.2 POE Input/Output Pins, changed
		857	23.2.7 Input Level Control/Status Register 3 (ICSR3), description of the OSTSTF flag, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	860	Figure 23.3 Low-Level Detection Operation, changed
		862	23.5.1 Transitions to Software Standby Mode or Deep Standby Mode, changed
		862	23.5.2 When POE is not Used, added
		862	23.5.3 Specifying Pins corresponding to the MTU, added
		24. 16-Bit Timer Pulse Unit (TPUa)	
		866	Figure 24.1 Block Diagram of TPU (Unit 0), changed
		867	Figure 24.2 Block Diagram of TPU (Unit 1), changed
		920	24.8 PPG Trigger, added
		927	24.10.2 Input Clock Restrictions, changed
		927	Figure 24.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode, changed
		25. Programmable Pulse Generator (PPG)	
		936	Figure 25.1 Block Diagram of PPG (Unit 0), changed
		936	Figure 25.2 Block Diagram of PPG (Unit 1), changed
		941	25.2.3 Output Data Registers H (PODRH) Output Data Registers L (PODRL) PPG0.PODRH description in the bit table changed, PODi description added PPG0.PODRL description in the bit table changed, PODi description added PPG1.PODRH description in the bit table changed, PODi description added PPG1.PODRL description in the bit table changed, PODi description added
		953	Figure 25.4 Schematic Diagram of PPG, changed
		958	Figure 25.9 Non-Overlapping Pulse Output, changed
		26. 8-Bit Timer (TMR)	
		All	Internal clock → frequency dividing clock, state → PCLK, changed
		966	Table 26.2 Pin Configuration of TMR, Cascaded connection, changed
		967	Figure 26.1 Block Diagram of TMR (Unit 0), changed
		968	Figure 26.2 Block Diagram of TMR (Unit 1), changed
		974	26.2.6 Timer Control/Status Register (TCSR) Bit table of TMR0.TCSR and TMR2.TCSR, Note 1, changed Bit table of TMR1.TCSR and TMR3.TCSR, Note 1, changed
		976	26.3.1 Pulse Output, changed
		978	Figure 26.7 Timing of Interrupt Flag Setting to 1 at Compare Match, changed
		980	Figure 26.10 Timing of Clearance by External Reset (Rising Edge), changed
		980	Figure 26.11 Timing of Clearance by External Reset (High Level), changed
		981	Figure 26.12 Timing of Overflow Interrupt Flag Setting to 1, changed
		982	Table 26.7 A/D Converter Activation, added
		983	Figure 26.13 Conflict between TCNT Write and Counter Clear, changed
		987	Figure 26.16 Continuous Output of Compare Match Interrupt Signal, changed
		27. Compare Match Timer (CMT)	
		All	Internal clock → frequency dividing clock
		988	Figure 27.1 Block Diagram of CMT (Unit 0), changed
		—	27.5.4 Point for Caution When Changing the Value of CMCR, deleted
		—	27.5.5 Point for Caution Regarding the Counter (CMCNT) and CMCOR, deleted
		28. Realtime Clock (RTCa)	
		All	Software reset → RTC software reset
		995	Table 28.1 Specifications of RTC, Interrupts changed
		996	Figure 28.1 Block Diagram of RTC, changed
		996	Table 28.2 Pin Configuration of RTC, description changed (RTCOUT),
		997	28.2 Register Descriptions, changed
		998	28.2.2 Second Counter (RSECCNT), value after reset changed, description of b7 in the bit table changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	998	28.2.3 Minute Counter (RMINCNT), value after reset changed, description of b7 in the bit table changed
		999	28.2.4 Hour Counter (RHRCNT), value after reset changed, description of b7 in the bit table changed
		1000	28.2.5 Day-of-Week Counter (RWKCNT), value after reset changed, description of b7 to b3 in the bit table changed
		1005	28.2.12 Day-of-Week Alarm Register (RWKAR), value after reset changed, description of b6 to b3 in the bit table changed
		1006	28.2.13 Date Alarm register (RDAYAR), value after reset changed, description of b6 in the bit table changed
		1007	28.2.14 Month Alarm Register (RMONAR), value after reset changed, description of b6 and b5 in the bit table changed
		1008	28.2.16 Year Alarm Enable Register (RYRAREN), value after reset changed, description of b6 to b0 in the bit table changed
		1009	28.2.17 RTC Control Register 1 (RCR1), description of bit table (AIE, CIE, PIE) changed, description of CIE bit changed
		1010	28.2.18 RTC Control Register 2 (RCR2), value after reset changed, bit name and description of RTCOE in the bit table changed, description of b7 (reserved bit) in the bit table changed, description of RTCOE and HR24 changed
		1012	28.2.19 RTC Control Register 3 (RCR3), description of RTCEN bit and b3 to b1 in the bit table changed, Note 1 added, register description changed, description of the RTCEN bit changed
		1015	28.2.22 Time Error Adjustment Register (RADJ), bit name of ADJ[5:0] changed, description of PMADJ[1:0] changed
		1016	28.2.23 Time Capture Control Register y (RTCCRy), description of the TCNF[1:0] and the TCEN changed
		1017	28.2.24 Second Capture Register y (RSECCPy), value after reset changed, description of b7 in the bit table changed
		1018	28.2.25 Minute Capture Register y (RMINCPy), value after reset changed, description of b7 in the bit table changed
		1018	28.2.26 Hour Capture Register y (RHRCPy), value after reset changed, description of b7 in the bit table changed
		1019	28.2.27 Date Capture Register y (RDAYCPy), value after reset changed, description of b7 and b6 in the bit table changed
		1020	Figure 28.2 Initial Settings after Power-On, changed
		1022	Figure 28.4 Setting the Time, changed
		1026	28.3.8.3 Procedure for Changing the Mode of Adjustment, changed
		1028	28.4 Interrupt Sources (1) and (3), changed
		1028	Table 28.4 RTC Interrupt Sources, changed, Note 1 added
		1030	Figure 28.13 Using Periodic Interrupt Function, changed
		1030	28.5.3 RTCOUT(1-Hz) Clock Output, changed
		1030	28.5.4 Transition to Low Power Consumption Modes after Setting Registers, changed
		1030	28.5.5 Points for Caution When Writing to and Reading from Registers, changed
		1031	28.5.6 Initialization Procedures When Realtime Clock Is Not in Use, added
		29. Watchdog Timer (WDTA)	
		1032	29.1 Overview, changed
		1032 to 1033	Table 29.1 Specifications of WDT, Auto-start mode, changed
		1033	Figure 29.1 Block Diagram of WDT, changed
		1041	29.3.1.2 Auto-Start Mode, changed
		1043	Figure 29.5 Control Waveforms Produced in Response to Writing to the WDTCR, changed
		1043	29.3.3 Refresh Operation [Sample sequences of writing that are not valid for refreshing the counter], changed
		1044	29.3.4 Status Flags, changed
		1045	29.3.5 Reset Output, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	1045	29.3.7 Reading Down-Counter Value, changed
		30. Independent Watchdog Timer (IWDTa)	
		1047	30.1 Overview, changed
		1047 to 1048	Table 30.1 Specifications of IWDT, Auto-start mode, changed
		1048	Figure 30.1 Block Diagram of IWDT, changed
		1056	30.3.1.2 Auto-Start Mode, changed
		1058	Figure 30.5 Control Waveforms Produced in Response to Writing to the IWDTCR, changed
		1058	30.3.3 Refresh Operation [Sample sequences of writing that are not valid for refreshing the counter], changed
		1060	30.3.4 Status Flags, changed
		1060	30.3.5 Reset Output, changed
		1061	30.3.7 Reading Down-Counter Value, changed
		1062	30.4.1 Refresh Operation, added
		31. Ethernet Controller (ETHERC)	
		1064	Table 31.2 Pin Configuration of ETHERC, Note 2, added
		1068	31.2.3 ETHERC Status Register (ECSR), description of BFR in the bit table changed, register description changed
		1070	31.2.5 PHY Interface Register (PIR), description of MDO in the bit table, changed
		1074	31.2.10 Automatic PAUSE Manual PAUSE Frame Register (MPR), register description, changed
		32. Ethernet Controller Direct Memory Access Controller (EDMAC)	
		1096	32.2.1 EDMAC Mode Register (EDMR), register description, changed
		1109	32.2.11 FIFO Depth Register (FDR), value after reset, changed
		33. USB 2.0 Host/Function Module (USBa)	
		1176	33.2.31 Pipe Cycle Control Register (PIPEPERI), description of IITV[2:0] in the bit table changed, description of IITV[2:0] and IFIS added
		1181	Table 33.8 Information Cleared by USB by Setting ACLRM = 1, changed
		1184	Table 33.10 Information Cleared by USB by Setting the ACLRM Bit = 1, changed
		1202	Table 33.12 Interrupt Sources, changed
		34. Serial Communications Interface (SCIc, SCId)	
		1242 to 1243	Table 34.1 Specifications of SCIc, Note 1, changed
		—	Table 34.2 Specifications of SCId, Note 1, deleted
		1245	Figure 34.1 Block Diagram of SCI0 to SCI4, SCI7 to SCI11, changed
		1246	Figure 34.2 Block Diagram of SCI5 and SCI6, changed
		1247	Figure 34.3 Block Diagram of SCI12 (SCId), changed
		1252 to 1254	34.2.5 Serial Mode Register (SMR), (2) Smart Card Interface Mode, description of PE in the bit table, changed
		1255 to 1258	34.2.6 Serial Control Register (SCR) (1) Serial Communications Interface Mode (SMIF in SCMR = 0), description of CKE[1:0] in the bit table changed, description of CKE[1:0] changed (2) Smart Card Interface Mode, description of CKE[1:0] in the bit table, changed
		1259 to 1262	34.2.7 Serial Status Register (SSR), (1) Serial Communications Interface Mode, description of TEND flag, changed (2) Smart Card Interface Mode, description of TEND flag, changed
		1263	34.2.8 Smart Card Mode Register (SCMR), Note 2 added, bit name of SINV and SDIR changed, description of SDIR changed
		1272 to 1273	34.2.10 Serial Extended Mode Register (SEMR), description of NFEN in the bit table changed, description of ACS0 and NFEN changed
		1274	34.2.11 Noise Filter Setting Register (SNFR), description of NFCS[2:0] in the bit table changed, description of NFCS[2:0] changed
		1280 to 1281	34.2.16 SPI Mode Register (SPMR), description of CKPOL and CKPH, changed
		1281 to 1282	34.2.17 Extended Serial Module Enable Register (ESMER), description of the ESME, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	1282	Table 34.26 Settings of the ESME Bits and Guaranteed Operation by Timer Operation Mode, added
		1284	34.2.20 Control Register 2 (CR2),description of the bit table (DFCS[2:0], BCCS[1:0] and RTS[1:0]) changed, note added
		1296	34.3.4 CTS and RTS Functions, changed
		1297	Figure 34.8 Sample SCI Initialization Flowchart (Asynchronous Mode), changed
		1298	34.3.6 Serial Data Transmission (Asynchronous Mode), changed
		1298	Figure 34.9 Example of Operation for Serial Transmission in Asynchronous Mode (from the Middle of Transmission until Transmission Completion), changed
		1299	Figure 34.10 Example of Serial Transmission Flowchart in Asynchronous Mode, changed
		1300	Figure 34.11 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (when RTS Function is not Used), changed
		1301	Figure 34.12 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (when RTS Function Is Used), added
		1302	Figure 34.13 Example of Serial Reception Flowchart (1) (Asynchronous Mode), changed
		1303	Figure 34.14 Example of Serial Reception Flowchart (2) (Asynchronous Mode), changed
		1305	Figure 34.16 Example of Multi-Processor Serial Transmission Flowchart, changed
		1307	Figure 34.18 Example of Multi-Processor Serial Reception Flowchart (1), changed
		1308	Figure 34.19 Example of Multi-Processor Serial Reception Flowchart (2), changed
		1309	34.5.CTS and RTS Functions, changed
		1310	Figure 34.21 S Example of SCI Initialization Flowchart (Clock Synchronous Mode), changed
		1311	34.5.4 Serial Data Transmission (Clock Synchronous Mode), changed
		1311	Figure 34.22 Example of Operation for Serial Transmission in Clock Synchronous Mode (from the Middle of Transmission until Transmission Completion), changed
		1312	Figure 34.23 Example of Serial Transmission Flowchart (Clock Synchronous Mode), changed
		1313	Figure 34.24 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (when RTS Function is not Used), changed
		1313	Figure 34.25 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (when RTS Function is Used), added
		1314	Figure 34.26 Example of Serial Reception Flowchart (Clock Synchronous Mode), changed
		1315	34.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode), changed
		1315	Figure 34.27 Example of Simultaneous Serial Transmission and Reception Flowchart (Clock Synchronous Mode), changed
		1319	34.6.5 Initialization of the SCI (Smart Card Interface Mode), changed
		1322	Figure 34.35 Sample Smart Card Interface Transmission Flowchart, changed
		1324	Figure 34.37 Sample Smart Card Interface Reception Flowchart, changed
		1324	34.6.8 Clock Output Control (1), (2) changed
		1328	34.7.2 Clock Synchronization, changed
		1330	Figure 34.45 Example of the Flow of SCI Initialization (for Simple I^2C Mode), changed
		1332	Figure 34.48 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (with Transmission Interrupts and Reception Interrupts in Use), changed
		1334	Figure 34.50 Example of the Procedure for Master Reception Operations in Simple I^2C Mode (with Transmission Interrupts and Reception Interrupts in Use), changed
		1335	34.8 Operation in Simple SPI Mode, changed
		1336	Table 34.29 States of Pins by Mode and Input Level on the SSn# Pin, Note 1 changed, Notes 2 and 3, added
		1337	34.8.5 SCI Initialization (Simple SPI Mode), changed
		1340	Figure 34.55 Sample Flowchart for Transmission of a Start Frame (1), changed
		1344	Figure 34.58 Sample Flowchart for Reception of a Start Frame (1), changed
		1348	34.9.4 Detection of Bus Collisions, changed
		1348	Figure 34.62 Example of Operations with Bus-Collision Detection, changed
		1349	Figure 34.63 Example of Operations with the Digital Filter, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	1351	34.9.7 Selectable Timing for Sampling Data Received through RXDX12, changed
		1351	Figure 34.65 Timing for Sampling of Data Received through RXDX12, changed
		1353	34.10 Noise Cancellation Function
		1354	34.11.2 Interrupts in Serial Communications Interface and Simple SPI Mode, changed
		1357	34.12.3 The Mark State and Production of Breaks, changed
		1361	34.12.11 Limitations on Simple SPI Mode, added
		1362	34.12.13 Limitation 2 on Usage of the Extended Serial Mode Control Section, changed
		1362	Figure 34.74 Example of Flowchart for Reception Error Handling (During Reception of the Start Frame), changed
		35. I^2C Bus Interface (RIIC)	
		1363	35.1 Overview, description of input level of the signals for RIIC, added
		1366	35.2.1 I^2C Bus Control Register 1 (ICCR1), Description of the bit table (ICE) changed, the ICE bit description changed
		1393	Figure 35.5 Example of RIIC Initialization Flow, changed
		1394	35.3.3 Master Transmit Operation (1), changed
		1397	35.3.4 Master Receive Operation (1), changed
		—	35.15.2 Setting Input Buffer Control Register, deleted
		36. CAN Module (CAN)	
		1432	Figure 36.1 Block Diagram of CAN Module, changed
		1433	36.2.1 Control Register (CTLR), description of the BOM[1:0], changed
		1447	36.2.8 Message Control Register j (MCTLj), description of the TRMREQ, changed
		1453 to 1455	36.2.11 FIFO Control Register (TFCR), description of the TFUST[2:0], changed
		1460	36.2.16 Channel Search Support Register (CSSR), register description, changed
		1460	Figure 36.4 Write and Read of CSSR and MSSR, changed
		1461	36.2.17 Acceptance Filter Support Register (AFSR), a note for the bit table, added
		1463 to 1465	36.2.19 Error Interrupt Factor Judge Register (EIFR), description of the EWIF flag, changed
		1470	Figure 36.9 Transition between CAN Operating Modes, Note 2, changed
		1472	Table 36.8 Operation in CAN Reset Mode and CAN Halt Mode, Note 4, changed
		1473	Figure 36.10 Sub-Modes of CAN Operation Mode, changed
		37. Serial Peripheral Interface (RSPI)	
		1485	Table 37.1 Specifications of RSPI, Others, changed
		1486	Figure 37.1 Block Diagram of RSPI, changed
		1488	37.2.1 RSPI Control Register (SPCR), description of the SPE, changed
		1490	37.2.3 RSPI Pin Control Register (SPPCR), SPOM, deleted (changed to a reserved bit)
		1492	37.2.5 RSPI Data Register (SPDR), register description, changed
		1493	Figure 37.2 Configuration of SPDR, added
		1494	Figure 37.3 Configuration of SPDR (Writing), added
		1495	Figure 37.4 Configuration of SPDR (Reading), added
		1497	37.2.7 RSPI Sequence Status Register (SPSSR), description and R/W status of b3 and b7 in the bit table, changed
		1499	37.2.9 RSPI Data Control Register (SPDCR), SLSEL[1:0] deleted (changed to a reserved bit), description of the bit table (SPFC[1:0]) changed, description of SPFC[1:0] and SPRDTD changed
		—	Figure 37.2 Frame Configurations and Examples of Combinations of Transmission and Reception Settings, deleted
		1500	Table 37.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits, changed
		1500	37.2.10 RSPI Clock Delay Register (SPCKD), register description, changed
		1501	37.2.11 RSPI Slave Select Negation Delay Register (SSLND), register description, changed
		1501	37.2.12 RSPI Next-Access Delay Register (SPND), register description, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	1506	Table 37.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode, changed
		1507	37.3.2 Controlling RSPI Pins, changed
		1507	Table 37.6 Relationship between Pin States and Bit Settings, changed
		1514	37.3.4 Data Format, changed
		1514	Figure 37.13 Outline of the Data Format (with Parity Disabled), added
		1514	Figure 37.14 Outline of the Data Format (with Parity Enabled), added
		1515	37.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0), changed
		1519	37.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1), changed
		1525	Figure 37.25 Operation Example of SPCR.TXMD = 0, changed
		1526	Figure 37.26 Operation Example of SPCR.TXMD = 1, changed
		1527	Figure 37.27 Operation Example of SPTI and SPRI Interrupts, changed
		1529	Table 37.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function, changed
		1530	Figure 37.28 Operation Example of OVRF Flag, changed
		1531	Figure 37.29 Operation Example of PERF Flag, changed
		1533 to 1540	37.3.10.1 Master Mode Operation (1) Starting a Serial Transfer, changed (2) Terminating a Serial Transfer, changed (3) Sequence Control, changed Figure 37.30 Procedure for Determining the Form of Serial Transmission in Master Mode, changed Figure 37.31 Concept of a Frame, changed Figure 37.32 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations, changed (4) Burst Transfer, changed (9) Flowchart of Operations, changed Figure 37.35 Example of Transfer Operation Flowchart in Master Mode (SPI Operation), changed
		1541 to 1543	37.3.10.2 Slave Mode Operation (2) Terminating a Serial Transfer, changed (6) Flowchart of Operations, changed Figure 34.37 Example Flowchart for Transfer Operations in Slave Mode (SPI Operation), changed
		1544 to 1548	37.3.12 Master Mode Operation (1) Starting a Serial Transfer, changed (2) Terminating a Serial Transfer, changed (3) Sequence Control, changed Figure 37.38 Procedure for Determining the Form of Serial Transmission in Master Mode, changed Figure 37.39 Concept of a Frame, changed Figure 37.40 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations, changed (4) Initialization Flowchart, changed Figure 37.41 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation), changed (5) Flowchart of Operations, changed Figure 37.42 Example Flowchart for Transfer Operations in Master Mode (Clock Synchronous Operation), changed
		1549 to 1551	37.3.13 Slave Mode Operation (4) Transfer Operation Flowchart, changed Figure 34.44 Example Flowchart for Transfer Operations in Slave Mode (CPHA = 1) (Clock Synchronous Operation), changed
		1554	Figure 37.48 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode), changed
		1556	37.3.17 Interrupt Sources, changed
		1556	37.4.2 Cautionary Note on the Low Power Consumption Functions, added
		38. IEBus Controller (IEB)	
		1557	Table 38.1 Specifications of IEB, Specifications of Operating frequency, changed
		1568	Figure 38.5 Block Diagram of IEB, changed
		1569	Table 38.7 Functions of Each Block, Bus interface, changed
		1587 to 1588	38.2.20 IEBus Clock Selection Register (IECKSR), description of CKS[2:0] in the bit table changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	39. CRC Calculator (CRC)	
		1607	39.2.1 CRC Control Register (CRCCR), description of DORCLR in the bit table changed, Note 1 added
		1608	39.2.3 CRC Data Output Register (CRCDOR), register description, changed
		40. 12-Bit A/D Converter (S12ADa)	
		All	Internal reference voltage → A/D internal reference voltage, single-cycle scan mode → single scan mode
		1613	Table 40.1 Specifications of 12-Bit A/D Converter, changed, Note 3 added
		1614	Table 40.2 Function Overview of the 12-Bit A/D Converter, changed, Note 3 added
		1615	Figure 40.1 Block Diagram of 12-Bit A/D Converter, changed
		1616	40.2.1 A/D Control Register (ADCSR), description of ADCS, changed,
		1617	Table 40.4 Scan Mode and Scope of Selectability for A/D Conversion, changed, description of ADST changed
		1621	40.2.7 A/D Control Extended Register (ADCER), Description of the ACE bit changed
		1622	Table 40.5 List of A/D Conversion Startup Sources, changed
		1627	40.2.13 A/D Sampling State Register 01 (ADSSTR01), added
		1628	40.2.14 A/D Sampling State Register 23 (ADSSTR23), Description of the bit table (b7 to b0), changed
		1632	40.3.4 Analog Input Sampling and Scan Conversion Time, changed
		1632	Table 40.6 Processing During a Scan Operation, changed
		1632	Table 40.7 Time Required for Scanning Operation, changed
		1633	40.3.5 Usage Example of ADDRy Register Automatic Clearing Function, changed
		1634	40.3.8 Starting A/D Converter with Synchronous Trigger from Peripheral Modules, changed
		1634	Table 40.8 A/D Conversion Start Sources to Be Selected, changed
		1635	40.3.8.2 Starting A/D Conversion using MTU TRGAN_0 and TPU TRGAN_1, changed
		1638	40.3.8.4 Starting A/D Conversion using MTU TRG4ABN_0 and TPU TRG4ABN_1, changed
		1640	40.5.2 Notes on Disabling A/D Conversion, changed
		1640	40.5.3 Notes on Restarting A/D Conversion Forced Stoppage, changed
		1640	40.5.5 Notes on Entering Low Power Consumption States, changed
		—	40.5.7 Allowable Impedance of Signal Source, deleted
		41. 10-Bit A/D Converter (ADb)	
		ALL	Activation → start
		1644	Table 41.2 Function Overview of the 10-Bit A/D Converter, Note 1 changed
		1645	Figure 41.1 Block Diagram of 10-Bit A/D Converter, changed
		1647	41.2.2 A/D Control/Status Register (ADCSR), Note 1 for the bit table, changed
		1655	Figure 41.5 Configuration Example of Extended Analog Input Circuit, changed
		1660	41.3.6 Using Synchronous Triggers from Peripheral Modules to Start A/D Conversion, added
		1660	41.3.6.2 Starting A/D Conversion Using MTU TRGAN_0 and TPU TRGAN_1, changed
		1666	41.6.2 Notes on Forcibly Stopping A/D Conversion, changed
		1666	41.6.3 A/D Conversion Restart Timing after Forced Stop, changed
		42. D/A Converter (DAa)	
		1671	Table 42.1 Specifications of D/A Converter, changed, Note 1 deleted
		1671	Figure 42.1 Block Diagram of D/A Converter, changed
		1673	42.2.2 D/A Control Register (DACR), register description changed, description of DAE, DAOE0 and DAOE1 changed
		1675	42.2.4 D/A A/D Synchronous Start Control Register (DAADSCR), Description of DAADST in the bit table changed, register description changed, description of DAADST changed
		1676	42.3 Operation 3, changed
		1678	42.4.5 Note on Usage when Measure against Interference between D/A and A/D Conversion is Enabled, changed

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	1678	42.4.6 Notes on Usage when Measure against Interference between D/A and A/D Conversion is Disabled, changed
		43. Temperature Sensor	
		1679	Figure 43.1 Block Diagram of Temperature Sensor, changed
		1681	43.3.1 Preparation of Using the Temperature Sensor, changed
		1682	43.3.2 Setting of 12-Bit A/D Converter, changed
		1683	Figure 43.2 Procedure for Using the Temperature Sensor, changed
		1684	Figure 43.3 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion, changed
		1684	Table 43.2 Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation, changed
		45. ROM (Flash Memory for Code Storage)	
		All	Writing/erasing → programming/erasure, mat → area
		1687	Figure 45.1 Block Diagram of ROM, changed
		1689	45.2.2 Flash Mode Register (FMODR), register description, changed
		1690	45.2.3 Flash Access Status Register (FASTAT), register description changed, Description of ROMAE changed
		1692	45.2.4 Flash Access Error Interrupt Enable Register (FAEINT), register description, changed
		1693	45.2.5 FCU RAM Enable Register (FCURAME), register description changed, description of FCRME changed
		1697	45.2.8 Flash Ready Interrupt Enable Register (FRDYIE), register description, changed
		1698 to 1700	45.2.9 Flash P/E Mode Entry Register (FENTRYR), register description, changed
		1701	45.2.11 Flash Reset Register (FRESETR), register description, changed
		1706	45.4 Block Configuration, changed
		—	Figure 45.4 Transitions between Operating Modes in Terms of the ROM, deleted
		1712	Table 45.8 Acceptable Commands and the State and Mode (ROM P/E Mode) of the FCU, changed
		1714	Figure 45.6 Procedure for Transition to ROM Read Mode, tFUCR, changed
		1715	Figure 45.8 Procedure for Transition to ROM Status Read Mode and the Status Checking, changed
		1718	Figure 45.11 Procedure for Firmware Transfer to FCU RAM, changed
		1720	Figure 45.12 Using the Peripheral Clock Notification Command, tFCUR, changed
		1722	Figure 45.13 Procedure for ROM Programming, tFCUR, changed
		1723	Figure 45.14 Procedure for ROM Erasure, tFCUR, changed
		1724	Figure 45.15 Procedure for Programming the Lock Bit, tFCUR, changed
		1725	Figure 45.16 Procedure for Reading Lock Bit in Register Read Mode, tFCUR, changed
		1728	Figure 45.18 Procedure for Programming/Erasure Suspension, tFCUR, changed
		1729	Figure 45.19 Procedure for Resuming Programming or Erasure, tFCUR, changed
		1730	Figure 45.20 Suspension during Programming, changed
		1734	Table 45.9 Error Protection Types (Types Dedicated to ROM and Types Common to ROM and E^2 DataFlash), Illegal command error, changed
		1741 to 1750	45.10.6 Inquiry/Selection Host Command Wait State (1) Supported Device Inquiry, changed (2) Device Selection, changed (9) Erasure Block Information Inquiry, changed (11) New Bit Rate Selection, the calculation formula for a bit rate error, changed
		1752 to 1759	45.10.8 Programming/Erasure Host Command Wait State, changed (128 bytes → 256 bytes) Table 42.16 Programming/Erasure Host Commands, changed (128 bytes → 256 bytes) Figure 42.30 Procedure for ROM Programming in Boot Mode, changed
		1754	(3) 256-Byte Programming, changed (128 bytes → 256 bytes)

Rev.	Date	Description	
		Page	Summary
0.90	Dec 12. 2011	1764 to 1765	45.14 Usage Notes (2) Suspending Programming or Erasure, changed (4) Reset during Programming or Erasure, changed (5) Prohibition of Non-Maskable Interrupts during Programming or Erasure, changed (7) Programming and Erasure in Low-Speed Operating Modes 1 and 2, added (8) Programming/Erasure Abnormal End, changed (9) Actions Prohibited during Programming and Erasure, changed (10) Notes on Flash Programming in Boot Mode or USB Boot Mode, added (11) Handling of the EXTAL Pin in Boot Mode, added (12) Handling of the EXTAL Pin in USB Boot Mode, added
		46. E2 DataFlash Memory (Flash Memory for Data Storage)	
		1766	Table 46.1 Specifications of E2 DataFlash Memory, Reading via the peripheral bus, changed
		1767	Figure 46.1 Block Diagram of E2 DataFlash Memory, changed
		1768	Table 46.2 Input and Output Pins Associated with the E2 DataFlash, changed (PC7 added)
		1775	46.2.6 E2 DataFlash Programming/Erasure Enable Register 0 (DFLWE0), register description, changed
		1776	46.2.7 E2 DataFlash Programming/Erasure Enable Register 1 (DFLWE1), register description, changed
		1777	46.2.8 Flash P/E Mode Entry Register (FENTRYR), register description changed, Note 2 changed
		1784	Table 46.5 FCU Command Formats (Commands Dedicated to the E2 DataFlash Memory), WA and BA, changed
		1785	Table 46.6 Acceptable Commands and the State and Mode (E2 DataFlash P/E Mode) of the FCU, changed
		1795	46.9 Usage Notes, (2) Reading in Low-Speed Operating Mode 2, added
		47. Boundary Scan	
		1796 to 1812	Added
		48. ROMless	
		1813 to 1823	Added
		Appendix 1. Port States in Each Processing Mode	
		1825 to 1827	Table 1.1 Port States in Each Processing State, changed
		Appendix 2. Package Dimensions	
		1828	Figure A. 177-pin TFLGA (PTLG0177KA-A), added
		1829	Figure B. 176-pin LFBGA (PLBG0176GA-A), added
		1831	Figure D. 145-pin TFLGA (PTLG0145KA-A), added
		1833	Figure F. 100-pin TFLGA (PTLG0100KA-A), added

RX63N Group, RX631 Group User's Manual: Hardware

Publication Date: Rev.0.50 May 12, 2011
Rev.0.90 Dec 12, 2011

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

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: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, 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 Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1

RX63N Group, RX631 Group

Renesas Electronics Corporation

R01UH0041EJ0090

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for renesas ***manufacturer:***

Other Similar products are found below :

EL4511CUZ-T7 PYB15-Q24-S5-H-U PQA30-D24-S24-DH PQA30-D48-S12-TH PYB30-Q24-T312-H-U PYB15-Q24-S5-H-T PYB15-Q24-S12-H-T V7815-500-SMT PYB20-Q48-S12-H-T PQZ6-Q24-S15-D PYB20-Q48-S5-H-T PYB20-Q24-S12-H-T VLED15-120-350 VGS-75-12 PYB15-Q24-S12-H-U VGS-50-15 VGS-50-24 R5F100GFAFB#V0 VGS-25-24 VGS-50-5 VGS-100-12 M30620FCAFP#U3 ETSA120500UD-P5P-SZ PDQ2-D24-S12-S PDS1-S12-D12-M PDS1-S12-D15-M PYB15-Q24-S12-T PYB15-Q24-S24 PYB20-Q24-S24-DIN PYB20-Q48-S12 R0K33062PS000BE R0K505220S000BE R0K561664S000BE R0K570865S000BE HC55185AIMZ R5F72145BDFA#V1 R5S72631P200FP R7S721001VCBGAC0 EPSA050250UB-P5P-EJ HS0005PUU01H IS82C55A-5 ISL29035EVAL1Z ISL55110IVZ ISL6730AEVAL1Z ISL68200DEMO1Z ISL78235EVAL2Z ISL78268EVAL1Z ISL91107IRA-EVZ ISL9220IRTZEVAL1Z ISL95870BIRZ